FIRIA LABS

Curriculum Guide

Mission Pack:

The Brain Decoded:

A Neural Network Adventure

Table of Contents

The Brain Decoded with CodeX Overview

This mission pack is an innovative cross-curricular educational toolkit designed to teach high school students both Python coding and the fascinating intricacies of the human brain. Through an engaging set of lessons, students will explore the inner workings of the human brain, discovering the connections between biological processes and coding principles.

During the missions, students learn about the brain, write programs as simulations of brain activity, and participate in activities that create mental models of how the brain works. The mission pack uses a combination of reading materials, CodeX programs, discussion, physical activities and written responses to create an immersive experience. No programming experience is required, but it is helpful.

Your Brain Decoded kit comes with a CodeX, peripherals (a NeoPixel Ring, a potentiometer and a 180 servo) and their connecting cables. The last mission includes an activity that optionally uses CodeX pods, which can be 3D printed: https://www.thingiverse.com/firia/designs

Mission 1: Brainstorm Bootcamp

During Brainstorm Bootcamp, students learn about CodeSpace and how it will guide them through the objectives with a text editor, objectives panel and goals. Students learn about helpful tools, like the toolbox, CodeTrek and hints. Basic Python programming for the CodeX is introduced, such as turning on pixels and displaying images. Students also use the peripherals that come with the kit: the NeoPixel Ring, 180 Servo, and the Potentiometer. As a final activity, students use radio signals to send messages to each other using the CodeX.

Mission 2: Neuron Navigator

Students learn about neurons and how the brain's inner signals guide everything we do. They will learn the parts of the neuron and how they work together to communicate. Students participate in a physical activity that forms a neural network. They also use the CodeX devices and programs to simulate a single neuron and a neural network.

Mission 3: Synaptic Sparks

During this mission students dive into the chemistry of the brain to see how neurotransmitters and neuromodulators affect mood, memory and behavior. First students learn about chemicals in the brain, like dopamine and serotonin, and how they regulate mood, memory and brain function. Then students use the CodeX device and programs to simulate different scenarios that affect brain activity and observe the results. The mission ends with a memory and reaction time test.

Mission 4: Language Logic

Students explore one of the brain's most fascinating skills: pattern recognition. First students read a short history on the roots of language. Then they use the CodeX device and programs to explore pattern recognition algorithms. Students consider the ethics of pattern recognition and how it is used in society. The mission ends with an English-Spanish partner madlibs activity.

Mission 5: Muscle Magic

During this mission students explore how motor neurons communicate with muscles. Topics covered are action potentials, the effect of neuron diameter, and the central pattern generator. Students use the CodeX device, the 180 servo and the potentiometer to simulate muscle activity and observe results. The mission ends with a "Go No-Go" brain training activity using the CodeX devices and the pods.

Planning and Pacing Guide

The Brain Decoded with CodeX Mission Pack includes five Missions.

- Mission 1 introduces CodeSpace and the CodeX device.
- Missions 2-5 each have a central theme for studying the human brain, and objectives that support the theme.
- All missions involve computer science.
- Missions 2-5 involve English, or Language Arts, extensively with reading materials, written responses, and reflection writing.
- The breakdown of subjects covered in each Mission is listed below. The main subject areas are mapped to standards. The cross-curricular activities are subject-specific, but they are not mapped to a specific standard.
- The suggested time gives time for the lessons in CodeSpace and a day for review. Including extensions and cross curricular activities will extend the amount of time needed.
- Other than Mission 1, each lesson is structured for a single 45-minute class period.
- Because the knowledge base and concepts build on each other from mission to mission, we recommend that the missions be completed in order. However, after Mission 1, they can be assigned out of order.

Mission 1: Brainstorm Bootcamp
Suggested Time: 3-4 days
Main Subject Areas:
Computer science
Cross Curricular Activities:
Language Arts (Obj 1, 2, 3, 4, 5, 6, 7)
Math (Obj 3, 4, 6, 7)
Physical Science (4, 5, 6, 7)

Mission 2: Neuron Navigator
Suggested Time: 6 days
Main Subject Areas:
Life Science (Biology)
Computer Science
English Language Arts
Cross Curricular Activities:
English Language Arts (Obj 1, 2, 5)
Math (Obj 3, 4)
Visual Arts (Obj 1, 2)

Mission 3: Synaptic Sparks
Suggested Time: 7 days
Main Subject Areas:
Physical Science
Computer Science
Engineering
Math
English Language Arts
Health / Physical Education
Cross Curricular Activities:
Math (Obj 2, 3, 4, 5, 6)
English Language Arts (Obj 1, 4, 5, 6)
Visual Arts (Obj 1)
Performing Arts (Obj 5)

Mission 4: Language Logic
Suggested Time: 7 days
Main Subject Areas:
English Language Arts
World Languages (Spanish)
Computer Science
Engineering
Cross Curricular Activities:
Math (Obj 2, 3, 4)
Social Studies/History (Obj 1, 5)
English Language Arts (Obj 1, 3, 5)
Physical Science (Obj 4)
Health (Obj 5)
World Languages (Obj 6)

Mission 5: Muscle Magic
Suggested Time: 7 days
Main Subject Areas:
Physical Science
Computer Science
English Language Arts
Math
Physical Education
Cross Curricular Activities:
Math (Obj 5, 6)
Social Studies/History (Obj 3)
English Language Arts (Obj 1, 4)
Physical Science (Obj 2)
Life Science (Obj 4, 6)
Visual Arts (Obj 1, 3)
Performing Arts (Obj 3)

Mission 1: Brainstorm Bootcamp

Overview:

This mission is all about becoming comfortable with the CodeSpace learning environment, the CodeX device, and Python programming. During Brainstorm Bootcamp, students learn about CodeSpace and how it will guide them through the objectives with a text editor, objectives panel and goals. Students learn about helpful tools, like the toolbox, CodeTrek and hints. Basic Python programming for the CodeX is introduced, such as turning on pixels and displaying images. Students also use the peripherals that come with the kit: the NeoPixel Ring, 180 Servo, and the Potentiometer. As a final activity, students use radio signals to send messages to each other using the CodeX.

Objectives:

Objective 1: Boot Up – This objective introduces the CodeSpace learning environment. They meet their guide Dr. Neuron Sparks.
Goal: Create a new file.

Objective 2: Boot Up II – Students learn about helpful tools in CodeSpace.
Goal: Click on a tool, CodeTrek and hints.

Objective 3: Introducing CodeX – Students connect the CodeX to the computer and CodeSpace.
Goal: Type in code and run their first program.

Objective 4: Code Action – Students use the buttons to light up pixels and display images.
Goal: Write a new program that uses buttons as input.

Objective 5: Plug In – This objective uses the NeoPixel Ring peripheral.
Goal: Write a program that lights up the pixels on the NeoPixel Ring.

Objective 6: Plug In... more! – This objective uses the 180 servo and potentiometer.
Goal: Write a program that uses the potentiometer to control the 180 servo.

Objective 7: Link Up – Students use the radio channel on their CodeX devices to send and receive messages to and from their classmates.
Goal: Write a program to send and receive radio messages.

Preparation and Materials:

- Create a class on the teacher dashboard.
- Students need a computer / laptop with the Chrome web browser.
- Make sure the students can successfully login to http://make.firialabs.com, create a student account and join the class with the code.
- Each student (or pair) needs a CodeX, peripherals and connecting cables.
- Supplemental materials found at learn.firialabs.com
  - Getting started with CodeSpace webpage – setting up a classroom, etc.
  - Getting Started in CodeSpace slide deck (for students)

Standards addressed in the mission:

Computer Science 9-12.CS.1, CS.2, AP.15, AP.16, AP.17

Mission Assessments:

Programs: first_program (Obj 3), button_fun (Obj 4), pixel_ring_fun (Obj 5), servo_fun (Obj 6), class_connect (Obj 7)
Brainstorm Bootcamp Mission Quiz

Mission 1: Brainstorm Bootcamp
Objective 1: Boot Up

Time Frame: 10 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I understand how to navigate CodeSpace.
- I can create a new file in CodeSpace.

Key Concepts

- CodeSpace has objectives, a text editor and goals for each objective.
- Goals must be completed before the next objective is unlocked.
- Files are saved automatically in the cloud. Use your account to access your program files.

Assessment Opportunities

- Create a new file in CodeSpace.
- Exit ticket on file names and PEP8 (in hints).

Success Criteria

- ☐ Identify the Objectives and Objectives Panel
- ☐ Identify and use the Text Editor
- ☐ Create a new file in CodeSpace

Standards

Computer Science: 9-12.CS.1 Describe ways in which abstractions hide the underlying implementation details of computing systems to simplify user experiences.

Vocabulary

- none

New Python Code

`# this is a comment`	Comment in code starts with #

Teacher Notes

- Extensions and cross-curricular projects are included to enhance the concepts in the objective. You can use the extensions to extend students' programming experience.
- The hints discuss saving files in CodeSpace. You can review program files with the students. It is especially important that students understand File-Save As. They will use this feature extensively.
- The hints discuss filenames and PEP8, which is a style guide for Python. You can have students look up the PEP8 style guide and go into more detail if you want.
- Another hint shows how to run the code. Students do not need to do this to meet the goal.

Extensions

- Include more than one comment in the program file.
- Have a discussion or practice worksheet on PEP8.
- Research or discuss "saving to the cloud."

Cross-Curricular

- **LANGUAGE ARTS:** Students write a summary of a topic covered in this objective.

Mission 1: Brainstorm Bootcamp
Objective 2: Boot Up II

Time Frame: 10 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can use helpful tools in CodeSpace.

Key Concepts

- CodeSpace has a toolbox for collecting "tools". The "tools" give CodeX and Python information to help with programming.
- CodeSpace has a programming guide called "CodeTrek" that walks through program code.
- CodeSpace includes Hints throughout the instructions to give additional information and helpful hints for the project.

Assessment Opportunities

- Click on a tool to add it to the toolbox.
- Click on CodeTrek and follow the instructions.
- Click on the Hints icon and read the hint.
- Exit ticket on when to use CodeTrek or Hints.
- Exit ticket on how to add a tool to the toolbox.

Success Criteria

- ☐ Tool added to toolbox
- ☐ CodeTrek followed
- ☐ Hint read

Standards

Computer Science: 9-12.CS.1 Describe ways in which abstractions hide the underlying implementation details of computing systems to simplify user experiences.

Vocabulary

- **Toolbox:** a collection of CodeX and Python concepts that can aid in programming. A tool must be clicked on and added to the toolbox in order to be accessed.
- **Debugging**: Fixing problems, or bugs, program code.

New Python Code

`from codex import *`	Added at the top of every program to access the codex library

Teacher Notes

- The tools, CodeTrek and Hints are embedded in CodeSpace as helpful tools to use during the mission pack. Students often overlook or don't use these tools. You may want to reinforce the need for using these tools to enhance their learning experience. Then check in occasionally to see if they are clicking on the tools, reading the information in CodeTrek and not just typing the code, and reading the hints.

Extensions

- If students didn't use the Hints or CodeTrek in Objective 1, have them go back and do so.
- Students can keep a log of tools they click on.
- Students can keep a journal of helpful hints they pick up along the way.

Cross-Curricular

- **LANGUAGE ARTS:** Students write a summary of debugging, which is a tool to click on.
- **LANGUAGE ARTS:** Have students explain to each other what the helpful tools are in CodeSpace, and when they should use each one.

Mission 1: Brainstorm Bootcamp
Objective 3: Introducing CodeX

Time Frame: 20-30 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can name the parts of the CodeX.
- I can connect the CodeX to a computer and CodeSpace.
- I can type code in a program file and run the code.

Key Concepts

- CodeX is a powerful embedded computer with loads of sensors, and input and output capabilities.
- Static electricity can harm the device.
- The CodeX needs to be connected to a computer and CodeSpace using a USB cable.
- When CodeX is connected to the computer, a window will pop-up. It can be closed; it is not needed for CodeSpace.
- When typing code, punctuation and indenting are VERY important! If a program won't run, check those two things first.

Assessment Opportunities

- Complete the program *first_program*.
- Exit ticket: Draw a picture of CodeX and label the parts.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors

Standards

Computer Science: 9-12.CS.2 Compare levels of abstraction and interactions between application software, system software, and hardware.

Vocabulary

- **Bitmap Image:** An image made up of graphics bits.

New Python Code

`from time import sleep`	Access to the sleep() function from the time library.
`display.show(pics.HEART)`	Displays a bitmap image from the pics Gallery.
`sleep(3)`	Program code pauses for the amount of seconds indicated.

Teacher Notes

- If you use Google Classroom, you can set up the class in CodeSpace so that students submit their programs through Google Classroom.
- The built-in bitmap images can be found in the pics gallery. Students should click on the **pics** tool so they can easily see the list of available images from the toolbox.
- When displaying a built-in bitmap image, the name must start with **pics.** in lower case, followed by the name of the image in ALL CAPS.

Extensions

- The program code has four images being displayed, with 3 seconds in between. Students can program more than four images.
- Have students vary the amount of time between images, and even have no time between images to see what happens.

Cross-Curricular

- **MATH:** Calculate the time it should take for the program to run. Then see if the predictions are correct.
- **LANGUAGE ARTS:** Have students write about one of the topics covered in this objective.

Mission 1: Brainstorm Bootcamp
Objective 4: Code Action

Time Frame: 10-30 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can control CodeX's four RGB pixel LEDs.
- I can use CodeX's six buttons for input.

Key Concepts

- The four RGB pixel LEDs are numbered 0, 1, 2 and 3. They can be set to any color.
- Students can use pre-defined colors for the RGB pixels without having to break down a color into Red/Green/Blue.
- The CodeX has 6 buttons for input.
- An "if" statement is used to create a branch of code that runs depending on a condition.

Assessment Opportunities

- Complete the program *button_fun*
- Exit ticket: Draw a picture of CodeX and label the four RGB pixel LEDs and the six buttons

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors

Standards

Computer Science: 9-12.AP.15 Iteratively design and develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

Vocabulary

- **NeoPixels:** LED lights that can display any color, using a combination of RED, GREEN and BLUE lights.
- **Boolean values:** Values that are either True or False (one of two states).

New Python Code

`while True:`	Infinite loop, keeps the code running continuously.
`if buttons.was_pressed(BTN_A):`	Used to create a branch of code that runs, depending on a condition. In this case, if Btn-A is pressed. If not pressed, the code is skipped.
`pixels.set(0, RED)`	Turns a single NeoPixel a color. The first argument is the pixel number, and the second argument is the color.

Teacher Notes

- The code inside an if statement must be indented! All lines of code need to have the same indentation. Otherwise an error will occur.
- The if statement must end with a colon (:). Otherwise an error will occur.
- A list of pre-defined colors can be found in the "RGB Colors" tool. Students should click on it and add it to their toolbox for easy reference. Pre-defined colors are typed in ALL CAPS.

Extensions

- The goal can be met by programming 4 of the buttons. Have students program all 6 buttons.
- The function "buttons.is_pressed()" is different from "buttons.was_pressed()". Have students try it and observe the difference.
- Have a lesson about RGB colors and have students define their own colors to use in the code using RGB.

Cross-Curricular

- **MATH:** RGB colors use values from 0-255, which is based on binary numbers. Have a lesson on binary numbers.
- **PHYSICAL SCIENCE:** Have a lesson on light and how red, green, and blue light can combine to create all the colors. Discuss how colors in light are different from colors in paint.
- **LANGUAGE ARTS:** Have students write about one of the topics covered in this objective.

Mission 1: Brainstorm Bootcamp
Objective 5: Plug In

Time Frame: 15-30 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can connect the NeoPixel Ring to the CodeX.
- I can control the pixels on the NeoPixel Ring.
- I can generate random integers within a specified range.

Key Concepts

- The NeoPixel Ring connects to the CodeX using one of the 4 black ports at the top.
- The NeoPixel ring can be connected using the wires, or plugged in directly to the port.
- The code to control the pixels on the NeoPixel ring is different from the code to control the pixels on the CodeX.

Assessment Opportunities

- Complete the program *pixel_ring_fun*.
- Exit ticket: Explain the three connecting wires: G, V and DI (or S).
- Check for Understanding: Create a diagram of how to connect the NeoPixel Ring.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors

Standards

Computer Science: 9-12.AP.15 Iteratively design and develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

Vocabulary

- **Peripheral:** Devices that interact with the CodeX, used for input and output.
- **NeoPixel Ring:** An output peripheral that has 8 RGB pixel LEDs arranged in a circle, numbered 0-7.

New Python Code

`from random import randint`	Access the randint() function from the random library.
`power.enable_periph_vcc(True)` `np = neopixel.NeoPixel(exp.PORT0, 8)`	Set up the NeoPixel Ring peripheral.

`RGB_RED = (20, 0, 0)`
`RGB_GREEN = (0, 20, 0)`
`RGB_BLUE = (0, 0, 20)`
	Define an RGB color, limiting the brightness to 20. The first number is the amount of RED, the second the amount of GREEN, and the third the amount of BLUE.

`def random_colors():`	Define a function, which is a block of code with a name that can be called anytime in the main program.

Teacher Notes

- The Hints give a lot of information about the NeoPixel ring and RGB colors. You may want to emphasize or review this information with your students.
- The code includes functions and for loops. If your students are familiar with coding, spend some time going through the code and understanding what each line does.

Extensions

- Students can define another color and program the B button to turn the pixels that color.
- Change the RGB colors defined at the top of the code to their own colors.

Cross-Curricular

- **PHYSICAL SCIENCE**: Have a lesson on light and its colors.
- **PHYSICAL SCIENCE**: Have a lesson on electricity, discussing ground and power connections.
- **LANGUAGE ARTS:** Have students write about one of the topics covered in this objective.

Mission 1: Brainstorm Bootcamp
Objective 6: Plug In...more!

Time Frame: 15-30 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can connect a servo to the CodeX.
- I can connect a potentiometer to the CodeX.
- I can control the servo by turning the potentiometer knob.

Key Concepts

- The servo and potentiometer are connected to the CodeX using connecting wires.
- The potentiometer needs the divider board to reach its full range of voltage.
- Attaching a horn to the servo helps you see its position and which way it is turning.

Assessment Opportunities

- Complete the program *servo_fun*.
- Exit ticket: Explain the purpose of the divider board.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors

Standards

Computer Science: 9-12.AP.16 Decompose problems into smaller subproblems through systematic analysis, using constructs such as procedures, modules, and/or classes.

Vocabulary

- **180 Positional Servo:** A motor that can rotate 180 degrees in each direction, but does not spin.
- **Potentiometer:** A knob that turns and varies the voltage; is used to control other peripherals.

New Python Code

`potentiometer = exp.analog_in(exp.PORT2)`	Set up the potentiometer

`PERIOD = 20`
`CYCLE = 2**16 // PERIOD`
`servo = exp.pwm_out(exp.PORT3, frequency=PERIOD)`
	Set up the 180 servo

`def set_servo(percent):` `  Return CYCLE * percent // 100`	Function that determines the position of the servo, determined by the percent
`reading = potentiometer.value`	Read the potentiometer's value

Teacher Notes

- The 180 servo uses PWM, and there is a lot of code and information about cycles and frequency. Students don't need to fully understand this to complete the goal or mission pack. However, this is an opportunity to dig deeper into electronics if you or the students are interested.
- Make sure that the peripherals are connected to the CodeX correctly. If the wires are reversed, they will not work properly. Students need to be gentle with the connections on the CodeX; the ports can bend.

Extensions

- Students can add additional elif statements to the code with their own potentiometer values and percents. Make sure to stay with the order of the statement, going from smallest value to largest.
- Turn the pixels a color in each branch as a visual clue to what the reading is.

Cross-Curricular

- **PHYSICAL SCIENCE**: Have a lesson on electronics and PWM.
- **MATH**: Look at the chart in the hints that shows the duty cycle for the servo. Make a graph from the chart.
- **LANGUAGE ARTS:** Have students write about one of the topics covered in this objective.

Mission 1: Brainstorm Bootcamp
Objective 7: Link Up

Time Frame: 15-30 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I turn on the CodeX radio and set a channel.
- I can send a message using a radio signal.
- I can receive a message using a radio signal.
- I can clear the CodeX LCD screen.

Key Concepts

- The CodeX has 14 channels. Any CodeX on the same channel can send and receive messages.
- If a CodeX is on channel 6 and another CodeX is on an adjacent channel, the messages can cross over.

Assessment Opportunities

- Complete the program *class_connect*.
- Exit ticket: Explain how to set up the radio, or send or receive messages.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors

Standards

Computer Science: 9-12.AP.17 Create computational artifacts using modular design.

Vocabulary

- **Radio Communication:** Point to point packet communication between multiple CodeX. All CodeX set to the same channel can send and receive messages with each other.

New Python Code

`import radio`	Access the functions from the radio library; must be the first instruction in the code
`radio.on()`	Turn on the radio
`radio.config(channel=6)`	Set the radio channel to 6
`radio.send(msg)`	Send the msg across the radio channel
`msg = radio.receive()`	Listen for a message from the radio channel
`if msg:`	If a message is received, run the block of code
`display.clear()`	Clear the LCD screen

Teacher Notes

- The code to import the radio module MUST be first. It sets up the memory.
- Students need to add their name to the message so you can know who is sending it. The instructions are in CodeTrek (step 2), but you may want to emphasize this.
- If you have a lot of students, this activity can get a little crazy. That is the fun of it! Students can clear their screen any time by pressing BTN_B.

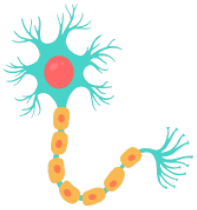
Extensions

- Add code to turn the pixels one color (or display an image) when sending a message and another when receiving a message or clearing the screen.
- For advanced students, CodeX connected to a computer can receive input from the keyboard. Students can customize the message sent by using an input() function.

Cross-Curricular

- **PHYSICAL SCIENCE**: Have a lesson on radio signals.
- **MATH**: Form a human chain and record the time it takes for students to send a message through the chain one at a time. Graph the results, and construct an equation.
- **LANGUAGE ARTS:** The objective starts with the inspiration for Dr. Neuron Sparks. Have students write about what inspires them.

Mission 2: Neuron Navigator



Overview:

This mission introduces the neuron. Students learn the part of a neuron and how they work together to communicate with each other. Students participate in a physical activity that represents a neural network, and then use the CodeX to model a neuron and simulate a neural network. Python programming concepts used are custom modules, functions, displaying text, global variables and parameters. The NeoPixel Ring is used during the neuron model program, and the radio signal is used during the neural network simulations. As a final activity, the class forms a neural network using the CodeX devices as neurons to send and receive signals.

Objectives:

Objective 1: Neuron Power – This objective introduces the neuron and explains the parts of a neuron. The objective involves reading the material, and no program is required.
Goal: Written response in a file.

Objective 2: Unplugged Chain – Students assume a role of neurons and participate in an activity that simulates a neural network. Students observe how each neuron plays a unique role in efficient brain communication.
Goal: Written reflection on physical activity in a file.

Objective 3: Code a Neuron – Students use the CodeX device to model a neuron. They write a program that cycles through the phases of neuron communication. The NeoPixel Ring is used during this program.
Goal: Program that models the phases of neuron communication.

Objective 4: Neural LinkUP – Students write a program to see how neurons talk to each other. They experience a digital network chain in action.
Goal: Program that uses the radio to send a signal to other CodeX in a chain of neural communication.

Objective 5: Network Test – This objective builds on the last objective by widening the communication from a single chain to a class-wide network.
Goals: Program that uses the radio to send and receive signals, simulating a neural network.

Preparation and Materials:

- Decks of cards (or slips of paper) for the unplugged activity
- Students need a computer / laptop with the Chrome web browser.
- Each student (or pair) needs a CodeX and connecting cables.
- Supplemental materials found at learn.firialabs.com
  - Objective 2 Activity Guide (one per student), Materials, Slides and Teacher Instructions
  - Objective 3 Activity Guide (one per student or programming pair)
  - Objective 4 Activity Guide (one per student or programming pair)
  - Objective 5 Activity Guide (one per student or programming pair)

Standards addressed in the mission:

Computer Science	9-12.AP.15, 9-12.AP.16, 9-12.AP.21
Life Science	LS1.A
ELA	WHST.9-10.2, WHST.11-12.2, WHST.9-10.4

Mission Assessments:

Written responses: neuron_power (Obj 1), unplugged_chain (Obj 2)
Programs: neuron_sim (Obj 3), neural_network1 (Obj 4), neural_network2 (Obj 5)
Neuron Navigator Mission Quiz

Mission 2: Neuron Navigator **Objective 1:** Neuron Power	**Time Frame:** 30-45 minutes (Suggested time for CodeSpace Lesson)
Learning Targets • I can identify the three parts of a neuron. • I can explain the responsibilities of each part of a neuron.	**Key Concepts** • Neurons are used in the messaging system of the brain. • A neuron has three basic parts, and each contributes to its communication.
Assessment Opportunities • Exit ticket: Draw a picture of a neuron and label its parts. • Written response that summarizes the neuron structure and how neurons transmit signals.	**Success Criteria** ☐ Use vocabulary from the objective ☐ Discuss the parts of a neuron and how they work together

Standards

Life Science:	LS1.A: Structure and Function Multicellular organisms have a hierarchical structural organization, in which any one system is made up of numerous parts and is itself a component of the next level. Science & Engineering Practices: Developing and Using Models Develop and use a model based on evidence to illustrate the relationships between systems or between components of a system. Crosscutting Concepts: Systems and System Models Models (e.g., physical, mathematical, computer models) can be used to simulate systems and interactions—including energy, matter, and information flows—within and between systems at different scales.
ELA:	WHST.9–10.2, WHST.11–12.2 Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes. WHST.9–10.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

Vocabulary

- **Neuron:** Fundamental building blocks of the brain; cells that transmit electrical and chemical signals.
- **Dendrites:** The input branches of a neuron that listen for signals from neighboring neurons.
- **Soma:** The neuron cell body that processes signals, deciding when to pass them along.
- **Axon:** The output wire of the neuron, sending messages to the next neuron in line.

Teacher Notes

- This objective is reading material for students. Feel free to supplement with any additional materials.
- The students are asked to create a file in CodeSpace for their written response. Students can complete a response in a different way, but students still need to create the file to meet the goal.

Extensions	**Cross-Curricular**
• Use supplemental materials to go more in-depth about neurons. • Have students create a video, web page, poster, or slides about neurons and give the presentation to an audience.	• **VISUAL ARTS:** Students can create an art project of a neuron (3D model, painting, drawing, etc.) • **LANGUAGE ARTS:** Students write a summary of a topic covered in this objective.

Mission 2: Neuron Navigator **Objective 2:** Unplugged Chain 	**Time Frame:** 45 minutes (Suggested time for CodeSpace Lesson)
Learning Targets • I can explain how neurons transmit signals. • I can explain how each neuron plays a unique role in brain communication. • I can use correct terms in a written response.	**Key Concepts** • Different types of neurons respond differently to signals. • The signal of a neuron must reach a strength threshold before it transmits to other neurons.
Assessment Opportunities • Record observations for each simulation • Turn in the Activity Guide • Written response that summarizes the communication of neurons	**Success Criteria** ☐ Use vocabulary from the objective ☐ Summarize the simulation and roles played ☐ Explain how neurons transmit signals

Standards

Life Science:	Science & Engineering Practices: Developing and Using Models Develop and use a model based on evidence to illustrate the relationships between systems or between components of a system.
ELA:	WHST.9–10.2, WHST.11–12.2: Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes.
Computer Science:	9-12.AP.21 Design and develop computational artifacts working in team roles using collaborative tools.

Vocabulary

- **Inhibitory Neuron:** A neuron that slows down or reverses a signal.
- **Malfunctioning Neuron:** A neuron that misses or skips a signal.

Teacher Notes

- The activity has supplemental materials to use:
  - A slide deck with narration to read to show the class that discusses the neural network and activity
  - Materials to prepare and print
  - An activity guide to print for each student
  - Teacher instructions to help you prepare for the activity
  - You will need decks of card, index cards, or slips of paper for the “neuron” messages
- The Activity Guide and Teacher Instructions give a lot of information and details about the simulation. Take time to read and go through them.
- Your class should run the simulation several times. Mix up the roles and scenarios each time.
- Give students time to reflect on or discuss each scenario before starting the next one.
- The students are asked to create a file in CodeSpace for their written response. Students can complete a response in a different way, but students still need to create the file to meet the goal.

Extensions	**Cross-Curricular**
• Use supplemental materials to go more in-depth about neurons. • Have students create a video, web page, poster, or slides about a neural network and give the presentation to an audience.	• **VISUAL ARTS:** Students can create a collaborative art project of a neural network. • **LANGUAGE ARTS:** Students write a summary of a topic covered in this objective.

Mission 2: Neuron Navigator
Objective 3: Code a Neuron

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can write a program that models a neuron.
- I can explain the phases of the program and each part of the neuron it represents.

Key Concepts

- The text reviews the parts of a neuron while explaining how it will be represented with the CodeX device.
- The CodeX program uses a custom module, which is a file students need to open and run first. This loads the file onto the CodeX. Once it is there, it will stay there and doesn't need to be loaded again.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *neuron_sim*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science: 9-12.AP.16 Decompose problems into smaller subproblems through systematic analysis, using constructs such as procedures, modules, and/or classes.

Vocabulary

- **Custom Module:** A file that contains a lot of Python code that will be used in the program. Once you run the file, it stays on the CodeX and its functions can be used anytime after being imported.

New Python Code

`from neurons import *`	Import the custom file to access it functions

`send_signal()`
`processing()`
`responding()`
	Function call

`break`	Exits the loop to stop the program

Teacher Notes

- The Objective has an Activity Guide for students. You can print the guide for each student or assign it digitally.
- The Hints give additional information about neurons and custom modules. You may want to emphasize this or review with the students.

Extensions

- Students who are familiar with coding can adjust some values in the code:
  - The weighted randint range in send_signal()
  - Any value added to or subtracted from strength
  - The value added to start_time

Cross-Curricular

- **MATH:** Students run their program several times and collect data. The data can be made into a graph and compared with other students' data. Use the graph to make inferences or draw conclusions, such as do students get faster with more trials? Develop an equation to describe the data.

Mission 2: Neuron Navigator
Objective 4: Neural LinkUP

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can collaborate in a team to form a neural chain using the CodeX devices.
- I can write a program that uses a radio signal to communicate with other CodeX devices.

Key Concepts

- The program uses a radio signal for communication.
- If the class is divided into multiple teams, assign each team a different nonadjacent radio channel.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *neural_network1.*

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science: 9-12.AP.21 Design and develop computational artifacts working in team roles using collaborative tools.

Vocabulary

- none

New Python Code

`CHAIN_NUM = 5`	Assign a value to a constant, designated by ALL CAPS. Constants are often found near the top of the code for easy access.
`my_role = 0`	Assign an initial value to a variable, designated by all lower case letters.

Teacher Notes

- The Hints give additional information about the activity. You may want to emphasize this or review with the students. It compliments the instructions in the activity guide.
- The Objective has an Activity Guide for students, which can be printed or assigned digitally.
- This is a single chain simulation. It can be run as a class and one long chain. If you use this option, the only change in code is for each student to put the number of participants (students in class) as the value for CHAIN_NUM.
- Or, you can place students in teams, and each team can form its own chain. If you divide the class into teams, each team needs to be on its own channel. Please assist teams in selecting channels. They should not be adjacent. For example, use even numbers or odd numbers. Then the changes in code are for the students to put the number of participants (students in the team) as the value for CHAIN_NUM and the number of their radio channel for TEAM_CHANNEL.
- Make sure that when students run the code and assign roles, that no role number is skipped.

Extensions

- Students who are familiar with coding can adjust some values in the code:
  - The weighted randint range in send_signal()
  - Any value added to or subtracted from strength
  - The value added to start_time

Cross-Curricular

- **MATH:** Teams run their simulation several times and collect data. The data can be made into a graph and compared with other teams' data. Use the graph to make inferences or draw conclusions, such as do teams get faster with more trials? Develop an equation to describe the data.

Mission 2: Neuron Navigator
Objective 5: Network Test

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can collaborate in a team to form a neural network using the CodeX devices.
- I can write a program that uses a radio signal to communicate with other CodeX devices.

Key Concepts

- The program uses a radio signal for communication.
- One CodeX is assigned role 1 and starts the communication. One CodeX is assigned the last role and ends the communication. All other roles can have multiple CodeX assigned.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *neural_network2*

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science:

9-12.AP.15 Iteratively design and develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

9-12.AP.16 Decompose problems into smaller subproblems through systematic analysis, using constructs such as procedures, modules, and/or classes.

Vocabulary

- none

New Python Code

`pixels.fill(BLUE)`	Light up all pixels on the CodeX the same color
`my_role = set_role()`	Function call when the function returns a value
`signal_received(pic_number)`	Function call that passes an argument to a parameter
`message = msg.split(',')`	Splits a string message into a list, using a comma to identify each list item.

Teacher Notes

- Assign roles before your students run the program. One student needs to start the communication and set their role to 1. One student needs to end the communication and set their role as the last one. All other students set their roles in between the first and last. There can be any number of CodeX devices assigned to each middle role, but there should be at least one CodeX assigned to each role.
- The Objective has an Activity Guide for students. You can print the guide for each student or assign it digitally.

Extensions

- Use the scenarios from Objective 2 for this digital simulation.

Cross-Curricular

- **LANGUAGE ARTS:** Compare and contrast this digital simulation with the physical simulation from Objective 2.

Mission 3: Synaptic Sparks

Overview:

This mission dives into the chemistry of the brain, including neurotransmitters and neuromodulators. Students learn how levels of dopamine and serotonin affect mood, memory and behavior. Students use the CodeX device to simulate scenarios and observe their effects. The NeoPixel Ring is used during the brain simulation program. As a final activity, students program and then participate in a memory and reaction tests.

Objectives:

Objective 1: Brain Chemicals – This objective introduces neurotransmitters and neuromodulators and gives students focus questions for the mission.
Goal: Written response in a file.

Objective 2: Mood and Memory – Students write a program that simulates many daily activities. They observe the effects of the activities on neuromodulators, memory and mood.
Goal: Program that simulates daily activities and their effects on mood and memory.

Objective 3: Your Brain on Sunshine – Students learn about the effects of sunshine and nature.
Goal: Program that simulates getting more or less sun, and going on nature walks.

Objective 4: Your Brain on Sleep – This objective discusses the sleep cycles and how sleep patterns affect brain activity.
Goal: Program that simulates sleep cycles and interruptions.

Objective 5: Your Brain on Drugs – Students learn how different types of drugs affect brain chemistry.
Goals: Program that simulates scenarios with different types of drugs and drug use.

Objective 6: Reaction Time! – Students review all the information from the mission and use the CodeX device for a culminating activity.
Goals: Program that gives a memory test followed by a reaction test, and then gives five different scenarios and how they can affect brain function.

Preparation and Materials:

- Flashlights for Objective 3
- Students need a computer, Chrome web browser, a CodeX, NeoPixel Ring and connecting cables.
- Supplemental materials found at learn.firialabs.com
  - Activity Guide (one per student or programming pair) for Objective 2, 3, 4, 5 and 6

Standards addressed in the mission:

Computer Science	9-12.AP.14, 9-12.AP.16, 9-12.AP.18
Health	1.1.A, 1.1.P, 1.2.A, 2.7.P, 5.2.M, 5.4.P
Physical Science	HS-PS3-1
Science	Science & Engineering Practices
Math	N-Q.1-3
ELA	ELA/Literacy RST.11-12.1, WHST.9-10.2
Physical Education	PE-HS1.1.3

Mission Assessments:

Written response: brain_chemicals (Obj 1)
Programs: brain_sim (Obj 2), sunshine (Obj 3), sleep_stages (Obj 4), drug_sim (Obj 5), reaction_time (Obj 6)
Synaptic Sparks Mission Quiz

Mission 3: Synaptic Sparks
Objective 1: Brain Chemicals

Time Frame: 30-45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain what a neuromodulator is.
- I can explain the difference between a neurotransmitter and a neuromodulator.

Key Concepts

- Dopamine and serotonin are powerful neurotransmitters and neuromodulators.

Assessment Opportunities

- Exit ticket: Make your own analogy of neurotransmitters and neuromodulators.
- Written response that summarizes the three prompt questions.

Success Criteria

- ☐ Discuss the impact of neuromodulators
- ☐ Give examples of neuromodulators influencing motivation, mood or memory
- ☐ What adjustments would you make to your symphony of the brain?

Standards

ELA: ELA/Literacy – RST.11–12.1 Cite specific textual evidence to support analysis of science and technical texts, attending to important distinctions the author makes and to any gaps or inconsistencies in the account. (HS-LS1-1), (HS-LS1-6)
WHST.9–10.2.a–f Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes. (HS-LS1-1), (HS-LS1-6)
WHST.11–12.2.a–e Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes. (HS-LS1-1), (HS-LS1-6)

Vocabulary

- **Neurotransmitter:** A single neuron that sends a specific targeted message to another neurotransmitter.
- **Neuromodulator:** Neurotransmitters working together to regulate how groups of neurotransmitters communicate.

Teacher Notes

- This objective is reading material for students. Feel free to supplement with any additional materials.
- The students are asked to create a file in CodeSpace for their written response. Students can complete a response in a different way, but students still need to create the file to meet the goal.

Extensions

- Use supplemental materials to go more in-depth about neurotransmitters and neuromodulators.
- Have students create a video, web page, poster, or slides about neuromodulators and give the presentation to an audience.

Cross-Curricular

- **VISUAL ARTS:** Students can create an art project to visualize neurotransmitters and neuromodulators.
- **LANGUAGE ARTS:** Students write a summary of a topic covered in this objective.

Mission 3: Synaptic Sparks
Objective 2: Mood and Memory

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can add code to a program.
- I can run a simulation of daily activities.
- I can observe and record the results of daily activities on brain function.

Key Concepts

- A variety of daily activities can affect brain function.
- This activity uses the CodeX device's built-in accelerometer.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *brain_sim.*

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science:	9-12.AP.14 Justify the selection of specific control structures by identifying tradeoffs associated with implementation, readability, and performance.
Health:	5.2.M Compare various coping mechanisms for managing stress.
Science:	Science & Engineering Practices: Constructing Explanations and Designing Solutions Construct an explanation based on valid and reliable evidence obtained from a variety of sources (including students' own investigations, models, theories, simulations, peer review) and the assumption that theories and laws that describe the natural world operate today as they did in the past and will continue to do so in the future.
Math:	N-Q.1-3: Reason quantitatively and use units to solve problems.

Vocabulary

- **Dopamine:** A neuromodulator that involves movement, memory, motivation, mood and learning.
- **Serotonin:** A neuromodulator that influences learning, memory and happiness. It also regulates sleep.

New Python Code

`life_events = ['Spend time', 'Listen to music']`	Define a list of strings
`index = random.randint(0, 9)`	Generate a random integer, including 0 and 9
`s = s + points//2`	// is floor, or integer, division, returning the integer only, no rounding

Teacher Notes

- The simulation uses the CodeX accelerometer as a form of exercise. It is built-in, so students don't need to do anything extra to connect it. The amount of shaking sets the value returned by the accelerometer.
- Students should follow the instructions on the activity guide to conduct the simulation. There are five different daily activities in the simulation. They can produce different results each time the button is pressed, so students should try each scenario several times.
- Students can repeat the simulation by starting it over, like going to a new day.

Extensions

- For students with experience in programming, they can add to the life events. Positive events need to be grouped together, and negative events together. They will also need to adjust the value index is compared to in event().
- Students can adjust the values in exercise() to vary the amount returned, or the number of shakes for each branch.

Cross-Curricular

- **MATH:** The program uses several built-in math functions, like abs(), int(), round() and min(). Discuss these functions, their differences, and instances when each one would be used.
- **MATH:** Look through the adjust_levels() function. Recreate it on paper and go through the calculations with data.

Mission 3: Synaptic Sparks
Objective 3: Your Brain on Sunshine

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain how sunshine and being in nature affect brain function.
- I can code and run a sunshine and nature walk simulation.
- I can observe and record the results of sunshine and nature walks on brain function.

Key Concepts

- Increased sunshine and nature walks have a positive effect on brain function.
- Lack of sun can negatively affect brain function.
- Students use a flashlight to simulate sunshine.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *sunshine*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Physical Science: HS-PS3-1. Create a computational model to calculate the change in the energy of one component in a system when the change in energy of the other component(s) and energy flows in and out of the system are known.

Vocabulary

- none

New Python Code

`reading = light.read()`	Read the light sensor and return its value
`if abs(reading - normal) > 350:`	The abs() function returns the absolute value of the argument. It can be used to detect a change, either positive or negative.
`mm = int(m_percent/100 * 255)`	The int() function returns the value of the calculation as an integer.
`m_percent = min(mm, 100)`	The min() function returns the smaller value

Teacher Notes

- The Hints include additional information about sunlight. You can review them with students.
- The simulation uses the CodeX light sensor. It is built-in, so students don't need to do anything extra to connect it. The amount of light shining on the light sensor is returned as a value.
- Students can use flashlights to add light, and cover the sensor with their hands to remove light.
- Students should follow the instructions on the activity guide to conduct the simulation. Six events for students to try are listed. They can be done in any order, and they can be repeated as often as desired.
- Students can repeat the simulation by starting it over, like going to a new day.

Extensions

- For students with experience in programming, they can adjust the light reading interval in sun_interval(). Or modify the code in sunlight().

Cross-Curricular

- **MATH:** The program uses several built-in math functions, like abs(), int(), and min(). Discuss these functions, their differences, and instances when each one would be used.
- **MATH:** Look through the adjust_levels() function. Recreate it on paper and go through the calculations with data.

Mission 3: Synaptic Sparks
Objective 4: Your Brain on Sleep

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets
- I can explain the sleep cycles.
- I can code and run a sleep simulation.
- I can observe and record the results of sleep on brain function.

Key Concepts
- Sleep cycles are N1, N2, N3 and REM.
- Uninterrupted sleep cycles have a positive effect on brain function.
- Not enough sleep or interruptions during sleep can negatively affect brain function.

Assessment Opportunities
- Turn in the Activity Guide.
- Complete the program *sleep_stages*.

Success Criteria
- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards
Health:
1.1.P Discuss the value of actively managing personal health behaviors (e.g., getting adequate sleep, practicing ergonomics, and performing self-examinations).
2.7.P Evaluate the need for rest, sleep, and exercise.

Vocabulary
- none

New Python Code

`small = []`	Define an empty list
`small.append("H"+str(hour))`	Add an item to the end of the list
`dis_big += 1`	Another way to increment a variable: dis_big = dis_big + 1
`hours = int(input('enter hours'))`	Type input on the console, convert to integer, and assign to a variable

Teacher Notes
- The Hints include a lot of additional information about sleep that students should read and review.
- The simulation uses the console for typing input. Students will open the console by clicking the Icon just below the Toolbox. You should see `>>>_` and a blinking cursor.
- Students should follow the instructions on the activity guide to conduct the simulation. Five different scenarios are suggested. Students can also try their own scenarios.

Extensions
- For students with experience in math and programming, look over the two lists and two functions for drawing webs. They can adjust the values to create their own webs.

Cross-Curricular
- **MATH:** The program uses a series of lines to form "webs". Research envelopes in math and have students try doing the math, or using the math to create their own webs with envelopes.
- **MATH:** Graph the data from the simulations. Make predictions. Develop an equation that explains the pattern.
- **LANGUAGE ARTS:** Students write about sleep cycles or how sleep affects brain function.

Mission 3: Synaptic Sparks
Objective 5: Your Brain on Drugs

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain how different types of drugs affect brain function.
- I can code and run a simulation on drug use and brain function.
- I can observe and record the results of a simulation on drug use.

Key Concepts

- Some types of drugs that affect brain function are stimulants, depressants and hallucinogens.
- Some drugs impact brain function positively, and other drugs have a negative impact.
- The program for this objective uses a custom module and the console for output.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *drug_sim*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Health:	1.1.A Describe the health benefits of abstaining from or discontinuing use of alcohol, tobacco, and other drugs. 1.2.A Explain the impact of alcohol, tobacco, and other drug use on brain chemistry, brain function, and behavior. 5.4.P Analyze how using alcohol, tobacco, and other drugs influences health and other behaviors.
Computer science:	9-12S.AP.18 Demonstrate code reuse by creating programming solutions using libraries and APIs. 9-12.AP.16 Decompose problems into smaller subproblems through systematic analysis, using constructs such as procedures, modules, and/or classes.

Vocabulary

- **Stimulants:** Drugs that increase dopamine and serotonin levels and speed up messages traveling between the brain and body.
- **Depressants:** Drugs that lower neuromodulator levels and decrease the electrical activity of brain cells.
- **Hallucinogens:** Drugs that disrupt communication between chemical networks throughout the brain.

New Python Code

- none

Teacher Notes

- The Hints include a lot of additional information about drug types that students should read and review.
- The simulation uses the console for most of the output. Students will open the console by clicking the Console Icon just below the Toolbox. All the results of each scenario will display on the Console.
- Students should follow the instructions on the activity guide to conduct the simulation. Five different scenarios are included in the simulation. The scenarios can be selected in any order. Students write the results of each scenario in their activity guide.

Extensions

- none

Cross-Curricular

- **MATH:** Graph the data from the simulations. Make predictions. Develop equations that explain the patterns.
- **LANGUAGE ARTS:** Watch a drug campaign video. Then come up with your own idea for a campaign video and write a script.
- **VISUAL ARTS:** Create an art project that depicts a healthy brain and a brain on drugs.
- **PERFORMING ARTS:** Act out a commercial for drug prevention.

Mission 3: Synaptic Sparks
Objective 6: Reaction Time!

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain how different brain states affect how quickly we respond.
- I can code and run a program that includes a memory test and reaction test.
- I can observe and record the results of the test with different scenarios.

Key Concepts

- Some factors that can positively affect brain function and response time are getting enough sleep, a dopamine boost, and sometimes stress.
- Some factors that inhibit brain function and response time are not getting enough sleep, sedation, and sometimes stress.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *reaction_time*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science:	9-12.AP.16 Decompose problems into smaller subproblems through systematic analysis, using constructs such as procedures, modules, and/or classes.
Physical Education:	PE-HS1.1.3 Identify, explain, and apply the skill-related components of balance, reaction time, agility, coordination, explosive power, and speed that enhance performance levels in aquatic, rhythms/dance, and individual and dual activities.

Vocabulary

- None

New Python Code

`cont = True`	Define a Boolean variable.
`while cont:`	Loop that continues while a Boolean variable is True.
`cont = ask_again()`	Assign a Boolean value to a variable from the return of a function.

Teacher Notes

- The Hints include a lot of additional information about the different brain states used in the program, and a short description of each scenario. Review this information with the students.
- The program for this objective uses a custom module, which contains most of the code.
- Students should follow the instructions on the activity guide to conduct the memory and reaction tests After running a test, students will select the five different scenarios and see how their score and time could be affected by different brain states. The scenarios can be selected in any order. Students write the results of each scenario in their activity guide.

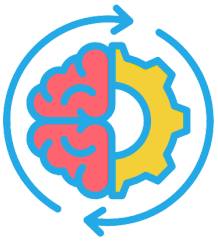
Extensions

- As a test for getting enough sleep, try this: People who do not get enough sleep have a lot of trouble remembering things. Run the memory and reaction test. Then ask the person to remember the order of the buttons from the memory test after the reaction test. See how much they remember. Record the results.

Cross-Curricular

- **MATH:** Graph the data from the simulations. Make predictions. Develop equations that explain the patterns.
- **LANGUAGE ARTS:** Select one of the scenarios. Write a story about a student in that scenario and how their memory and reaction times are affected.

Mission 4: Language Logic



Overview:

In this mission we explore pattern recognition and how it relates to language acquisition. Students will discover how the human brain looks for patterns in spoken and written language. They learn about searching algorithms and use the CodeX devices and programs to search for patterns in words, grids, and sound waves. Students also learn about and consider the ethics of using pattern recognition in real-world applications. The final activity is to use English and Spanish words in a partner-madlibs story.

Objectives:

Objective 1: Roots of Language – This objective discusses the evolution of human communication. Students trace the history of language from gestures and facial expressions to a shared proto-language.
Goal: Written response in a file.

Objective 2: Pattern Recognition – Students learn about the parts of the brain that work together to make sense of patterns in everything from faces to language. The concept of algorithm is introduced.
Goal: Program that looks for consecutive repeated letters in a word.

Objective 3: Pattern Detection – Students learn about an advanced search algorithm for detecting patterns.
Goal: Program that searches for a pattern in a grid of characters.

Objective 4: Language Patterns – This objective looks at patterns in spoken language..
Goal: Program that uses pattern detection to search a sound wave (speech) for syllables.

Objective 5: Ethical Considerations – Students research ways that pattern recognition is used in real-world applications and address ethical considerations.
Goals: Two written responses in files.

Objective 6: Language Acquisition – Students learn about areas in the brain that process language. They learn about language acquisition by using patterns and rules.
Goals: Program that has a student input words and uses them in a partner-swap madlibs story.

Preparation and Materials:

- Students need a computer, Chrome web browser, a CodeX and connecting cable.
- Supplemental materials found at learn.firialabs.com
  - Activity Guide (one per student or programming pair) for Objective 2
  - Activity Guide (one per student or programming pair) for Objective 3
  - Activity Guide (one per student or programming pair) for Objective 4
  - Activity Guide (one per student or programming pair) for Objective 6

Standards addressed in the mission:

Computer Science	9-12.DA.8, 9-12.DA.10, 9-12.AP.14, 9-12.IC.24, 9-12.IC.28, 9-12.IC.30
Engineering Design	HS-ETS1-1
ELA	W.9-10.1, W.11-12.1, RST.9-10.4, RST.11-12.4, WHST.9-10.2.a-f, WHST.11-12.2.a-e, HS-LS1-1, HS-LS1-6, L.11-12.3, L.9-10.4, L.11-12.4
World Languages	WL.CM5.N

Mission Assessments:

Written response: language_roots (Obj 1), ethics_prompt1 (Obj 5), ethics_prompt2 (Obj 5)
Programs: repeated_letters (Obj 2), pattern_detect (Obj 3), syllables (Obj 4), partner_madlibs (Obj 6)
Language Logic Mission Quiz

Mission 4: Language Logic **Objective 1:** Roots of Language	**Time Frame:** 30-45 minutes (Suggested time for CodeSpace Lesson)
Learning Targets • I can explain the evolution of human communication.	**Key Concepts** • Recognizing patterns is a key to understanding language.
Assessment Opportunities • Exit ticket: Give an example of patterns in language. • Written response reflection.	**Success Criteria** ☐ Summarize reading material ☐ Give examples of modern words or patterns that might survive for millinia, and why.

Standards

ELA:	W.9-10.1 W.11-12.1 Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
Computer Science:	9-12S.IC.28 Evaluate how computational innovations that have revolutionized aspects of our culture might evolve.
Engineering Design:	HS-ETS1-1. Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.

Vocabulary
- None

Teacher Notes
- This objective is reading material for students. Feel free to supplement with any additional materials.
- The students are asked to create a file in CodeSpace for their written response. Students can complete a response in a different way, but students still need to create the file to meet the goal.

Extensions
- Use supplemental materials to go more in-depth about the history of language.
- Have students create a video, web page, poster, or slides about neuromodulators and give the presentation to an audience.

Cross-Curricular
- **SOCIAL STUDIES/HISTORY:** Students can research the development of language and/or communication in a country or region. What influenced the development, and how did it impact culture?
- **SOCIAL STUDIES:** Compare and contrast languages from different countries. What patterns do they have? How does their language affect culture?
- **LANGUAGE ARTS:** Students write a summary of a topic covered in this objective.

Mission 4: Language Logic
Objective 2: Pattern Recognition

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can identify parts of the brain used in pattern recognition.
- I can explain what an algorithm is.

Key Concepts

- Many parts of the brain work together to recognize, remember and make sense of patterns in everything from faces to languages.
- Computers can recognize patterns using algorithms.
- Using an algorithm and neural network processing are different, but learning about algorithms is a good place to start.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *repeated_letters*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science	9-12.AP.14 Justify the selection of specific control structures by identifying tradeoffs associated with implementation, readability, and performance.
ELA:	RST.9–10.4 RST.11–12.4 Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 9–10 texts and topics.

Vocabulary

- **Algorithm:** A step-by-step process for solving problems or performing calculations.
- **Visual Cortex:** The part of the brain that processes visual information.

New Python Code

`for i in range(len(word))` `for i in repeated_letters`	Looping structure that traverses a list (word). Looping structure that traverses a list (repeated_letters).
`word = input("enter a word")`	Type string input to the console and assign it to a variable.
`if word[i] == word[i+1]:`	If statement that compares the current item in the list with the next item in the list.
`display.print(word[i], end='')`	Prints the current item in the list and then stays on the line; it does not go to the next line for the next item.

Teacher Notes

- The program uses the Console for input. Students need to click on the Console Icon below the Toolbox. When the program runs, a prompt will be displayed. Type the word to search and press <Enter>.
- Students should follow the instructions on the activity guide, and record their results.

Extensions

- For students with experience in programming, have them review the code for find_repeating_letters() and print_word(). Explain each line of code. How are they alike? How are they different?

Cross-Curricular

- **MATH:** The field of mathematics uses algorithms, just like computers. One interesting algorithm is Kaprekar's Constant. Follow the algorithm with different numbers.

Mission 4: Language Logic
Objective 3: Pattern Detection

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain feature extraction and what it is used for.
- I can explain data representation for pattern detection in an algorithm.
- I can explain data representation in a neural network.

Key Concepts

- Pattern recognition using an algorithm involves searching through rows and columns of data to match patterns.
- Brains and neural networks use parallel processing and interconnected layers of neurons to recognize patterns.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *pattern_detect*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science: 9-12.DA.8 Translate between different representations of data abstractions of real-world phenomena, such as characters, numbers, and images.

Vocabulary

- **Feature extraction:** The process of looking for specific features in data.
- **Matrix:** A list of lists, a two-dimensional (or more) grid of values.
- **Vector:** A one-dimensional list of values.

New Python Code

`for i in range(len(grid)):` `  for j in range(len(grid[i]):`	Looping structure that traverses a grid (outside loop is each row, and inside loop is each column).

Teacher Notes

- Typing in the code for the program can be tricky. Sometimes parenthesis () are used, and sometimes square brackets []. Encourage students to read the CodeTrek slowly and carefully, and to pay extra attention as they are typing code.
- Students should follow the instructions on the activity guide, and record their results.

Extensions

- The activity guide asks students to change a pattern and then either change or add a grid. Encourage students to several grids, and make them different sizes. They do not need to be a square shape.
- Four patterns are used since there are four buttons. Change the code to add more patterns.
- As an advanced challenge, students can change the pattern to something other than a 2x2 grid, but that requires changing the search algorithm. Challenge them to do this.

Cross-Curricular

- **MATH:** This program uses an advanced search to iterate over rows and columns. It expands the original search algorithm. Think of a simple algorithm in math and how it can be expanded to a more advanced algorithm. Example: adding two single-digit numbers can be expanded to adding two-digit numbers.
- **LANGUAGE ARTS:** Practice technical writing by composing a manual article on how the searching algorithm program works.

Mission 4: Language Logic
Objective 4: Language Patterns

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain feature extraction to simplify sound.
- I can use the CodeX device to record sound and analyze the peaks and gaps.

Key Concepts

- Your brain doesn't analyze every tiny detail of sound waves. It zooms in on the essentials.
- A pattern detection algorithm can search for peaks and gaps in a sound wave to detect syllables.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *syllables*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science: 9-12.DA.10 Create data visualizations to help others better understand real-world phenomena.

ELA: RST.9–10.4 RST.11–12.4 Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 9–10 texts and topics.

Vocabulary

- **Envelope:** In audio signal processing, an envelope represents the overall shape of a sound signal's volume over time, leaving out the details of the sound's frequency content. It describes how the intensity of the sound changes.

New Python Code

`x % 4`	% is the symbol for modulo division, which returns the integer remainder of division. In this example, possible values would be limited to 0, 1, 2, or 3.

Teacher Notes

- The Hints and CodeTrek explain the code. Students can go through the hints and CodeTrek carefully, and review with each other, to understand what the code does and how it works.
- The program requires students to use the built-in microphone on the CodeX. It is located just to the right of BTN-R. Students will need to get up close and speak loudly for best results.
- Getting the right values for the constants is tricky. Read the hints for best results, and try them first. If the program is still not detecting syllables, students can adjust the values of THRESH and MIN_GAP. Try THRESH first. If the sound wave isn't reaching the red markers, make the THRESH value smaller. Then adjust the MIN_GAP, or the gaps in the sound, to get the correct number of syllables.
- The program won't count all the syllables every time for every word. Some words have more peaks and gaps than others. This can be a point of discussion.
- Students should follow the instructions on the activity guide, and record their results.

Extensions

- This program uses similar searching algorithms as the programs in Objective 2 and Objective 3. Compare and contrast the code that searches for patterns in each of the programs.

Cross-Curricular

- **MATH:** This program uses modulo division. Have a lesson on integer and modulo division and give practice problems.
- **SCIENCE:** Add a lesson on what sound waves are and how they are created. Look for physical examples, or try different experiments.

Mission 4: Language Logic
Objective 5: Ethical Considerations

Time Frame: 30-45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain several ways pattern recognition is used in real-world applications.
- I can discuss ethical considerations with using pattern recognition in real-world applications.
- I can discuss bias and privacy issues with pattern recognition used in real-world applications.

Key Concepts

- Biometrics use pattern recognition to identify individuals.
- Pattern recognition can be used in surveillance systems.
- Data analysis uses pattern recognition to process vast amounts of data, looking for trends.

Assessment Opportunities

- Written response reflection for prompt 1.
- Written response reflection for prompt 2.

Success Criteria

- ☐ Use examples from the reading material to support key points in a written response.
- ☐ Consider ethical principals of pattern recognition use in a written response.
- ☐ Discuss pattern recognition bias in a written response.

Standards

ELA:	WHST.9–10.2.a–f WHST.11–12.2.a–e Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes. (HS-LS1-1), (HS-LS1-6)
Computer Science:	9-12.IC.24 Identify impacts of bias and equity deficit on design and implementation of computational artifacts and apply appropriate processes for evaluating issues of bias. 9-12.IC.30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.

Vocabulary

- **Biometrics:** The use of physical or behavioral characteristics to identify individuals.
- **Surveillance:** The monitoring of behavior or activities.
- **Data Analysis:** The process of examining data to extract useful information and make conclusions.

Teacher Notes

- This objective is reading material for students. Feel free to supplement with any additional materials.
- The students are asked to create two files in CodeSpace for their written responses. Students can complete their responses in a different way, but students still need to create the files to meet the goals.

Extensions

- Use supplemental materials to go more in-depth about the history of language.
- Have a class debate on the ethical considerations and technology.
- Have students create a video, web page, poster, or slides about ethical considerations and give the presentation to an audience.

Cross-Curricular

- **SOCIAL STUDIES:** Select a country and examine its society in relation to ethical use of technology. What is accepted, and what is considered unethical?
- **HISTORY:** Research the history of biometrics, data analysis or surveillance. How has it been changed by technology?
- **HEALTH:** Research how technology and pattern recognition has affected the health field or how it is used to help people live healthy lives.
- **LANGUAGE ARTS:** Students write a summary of a topic covered in this objective.

Mission 4: Language Logic
Objective 6: Language Acquisition

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can identify the areas of the brain that are responsible for processing language.
- I can explain the use of patterns in language.
- I can explain the importance of repetition when learning something new.

Key Concepts

- Our brain looks for patterns in language for clear communication.
- Even when words in a sentence are out of order, our brains can recognize familiar patterns and extract meaning.
- Each encounter with familiar patterns strengthens the neural connections.

Assessment Opportunities

- Exit ticket: Identify and briefly explain the areas of the brain responsible for language.
- Turn in the Activity Guide.
- Complete the program *partner_madlibs*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

World Languages: WL.CM5.N Demonstrate understanding of words, phrases (signs & fingerspelling in ASL), and simple sentences on very familiar common daily topics. Use orthography, phonology, ASL parameters, and very basic sentence-level elements (morphology and/or syntax).
WL.CM7.I Identify similarities and differences in the basic sentence-level elements (morphology and syntax) of the languages known.

ELA: L.11-12.3 Apply knowledge of language to understand how language functions in different contexts, to make effective choices for meaning or style, and to comprehend more fully when reading or listening.
L.9-10.4 L.11-12.4 Determine or clarify the meaning of unknown and multiple-meaning words and phrases based on grades 9–10/ 11–12 reading and content, choosing flexibly from a range of strategies.

Vocabulary

- **Wernicke's area:** The area in the brain responsible for understanding words and their meanings.
- **Broca's area:** The area in the brain responsible for speech production and grammatical structures.

New Python Code

`print("As they "+ verb1 +" through")`	Print a string with a variable. The + concatenates the strings.
`noun=input('enter noun').upper()`	Converts the inputted string to ALL CAPS

Teacher Notes

- The program for this objective requires students to work in pairs. It will not work individually, or correctly with a group of three. If you have an odd number of students, either partner with a student or have a group of three work two at a time. They can switch partners and run the code again if necessary.
- The program uses the Console extensively – both for inputting words and displaying the madlib story. The CodeX devices will need to remain connected to a computer while the program is running.
- Students should follow the instructions on the activity guide, and record their results.

Extensions

- This activity guide includes extensions for adding to the program.

Cross-Curricular

- **WORLD LANGUAGES:** Compose a short story for madlibs in a different language.

Mission 5: Muscle Magic

Overview:

During this mission students explore how motor neurons communicate with muscles. Topics covered are action potentials, the effect of neuron diameter, and the central pattern generator. Students use the CodeX device, the 180 servo and the potentiometer to simulate muscle activity and observe results. The mission ends with a “Go No-Go” brain training activity using the CodeX devices and the pods.

Objectives:

Objective 1: Motor Neurons – This objective discusses motor neurons and communication with muscles.
Goal: Written response in a file.

Objective 2: Motor Neuron Signals – Students learn about action potentials and how the diameter of a motor neuron impacts the speed of the signals.
Goal: Program that demonstrates the relationship between neuron diameter and action potential speed.

Objective 3: Neural Rhythms – Students learn about the rhythm generator circuit and rhythms created when synchronized neurons take turns firing.
Goal: Program that simulates a neural rhythm pattern.

Objective 4: Muscle Response – This objective discusses reflexes and the central pattern generator.
Goal: Program that simulates chewing and the CPG that sets it in motion.

Objective 5: Athletic Reaction – Students learn about motor neuron recruitment during different activities..
Goal: Program that generates a graph depicting motor neuron recruitment for activities of different speeds.

Objective 6: Reaction Revolution – This objective summarizes the learning in the mission pack and discusses training your brain to focus and perform better through practice.
Goal: Program that trains your brain to focus and perform during “Go No-Go” practice sessions.

Preparation and Materials:

- Students need a computer, Chrome web browser, a CodeX device, the potentiometer and 180 servo.
- Optional: 3D printed CodeX pods https://www.thingiverse.com/firia/designs
- Supplemental materials found at learn.firialabs.com
  - Activity Guide (one per student or programming pair) for Objective 2
  - Activity Guide (one per student or programming pair) for Objective 3
  - Activity Guide (one per student or programming pair) for Objective 4
  - Activity Guide (one per student or programming pair) for Objective 5
  - Activity Guide (one per student or programming pair) for Objective 6

Standards addressed in the mission:

Computer Science	9-12.DA.10, 9-12.AP.12, 9-12.AP.13, 9-12.AP.14
Physical Science	LS1.A, HS-LS1-2, HS-PS4-5, Science & Engineering Practices, Crosscutting Concepts
ELA	RST.11-12.1, RST.11-12.2
Math	S-ID.2
PE	PE-HS1.1.3

Mission Assessments:

Written response: motor_neurons (Obj 1)
Programs: action_potentials (Obj 2), neural_rhythms (Obj 3), chewing_sim (Obj 4), athlete_reaction (Ojb 5), train_your_brain (Obj 6)
Language Muscle Magic Mission Quiz

Mission 5: Muscle Magic
Objective 1: Motor Neurons

Time Frame: 30-45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain what a motor neuron does.
- I can explain how action potentials work.

Key Concepts

- The brain talks to muscles using motor neurons.
- Signals that travel through motor neurons are called action potentials.
- When the action potential arrives at the neuromuscular junction, it triggers a synaptic transmission and a contraction happens.

Assessment Opportunities

- Exit ticket: Recreate the motor neuron diagram.
- Written response reflection.

Success Criteria

- ☐ Summarize the motor neuron diagram.

Standards

Physical Science:	HS-LS1-2 Develop and use a model to illustrate the hierarchical organization of interacting systems that provide specific functions within multicellular organisms. LS1.A: Structure and Function Multicellular organisms have a hierarchical structural organization, in which any one system is made up of numerous parts and is itself a component of the next level. Science & Engineering Practices: Developing and Using Models Develop and use a model based on evidence to illustrate the relationships between systems or between components of a system. Crosscutting Concepts: Systems and System Models Models (e.g., physical, mathematical, computer models) can be used to simulate systems and interactions—including energy, matter, and information flows—within and between systems at different scales.
ELA:	RST.11–12.2 Determine the central ideas or conclusions of a text; summarize complex concepts, processes, or information presented in a text by paraphrasing them in simpler but still accurate terms. RST.11-12.1: Cite specific textual evidence to support analysis of science and technical texts, attending to important distinctions the author makes and to any gaps or inconsistencies in the account.

Vocabulary

- **Motor neuron:** Special nerve cells that carry orders from the brain or spinal cord directly to your muscles.
- **Action potentials:** Electrical impulses, or signals, generated by the brain and transmitted by motor neurons.
- **Neuromuscular junction:** The meeting point between a muscle and a motor neuron where the action potentials are transmitted.

Teacher Notes

- This objective is reading material for students. Feel free to supplement with any additional materials.
- The students are asked to create a file in CodeSpace for their written response. Students can complete a response in a different way, but students still need to create the file to meet the goal.

Extensions

- Use supplemental materials to go more in-depth about the history of language.
- Have students create a video, web page, poster, or slides about motor neurons and give the presentation to an audience.

Cross-Curricular

- **LANGUAGE ARTS:** Students write a summary of a topic covered in this objective.
- **VISUAL ARTS:** Students can create an art project or 3D model of a motor neuron.

Mission 5: Muscle Magic
Objective 2: Motor Neuron Signals

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain how the diameter of a motor neuron impacts the speed of action potentials.
- I can use variables and branching in a program.

Key Concepts

- The diameter of the motor neuron impacts the speed of action potentials moving down neurons.
- A neuron with a wider diameter increases action potential velocity.
- Variables and branching are key components of programming.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *action_potentials*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science	9-12.AP.14 Justify the selection of specific control structures by identifying tradeoffs associated with implementation, readability, and performance.
Math:	S-ID.2 Summarize, represent, and interpret data on a single count or measurement variable.

Vocabulary

- **Variable:** A name for a value that can change during the program, making it easier to work with data.
- **Branching:** A programming structure that enables one block of code to run, based on a condition.

New Python Code

`def muscle_speed(delay, angle):`	Define a function with parameters. Parameters receive their values from a function call's arguments.
`global forward, percent`	The global command allows global variables to be updated, or changed, in a function.
`time.sleep_ms(delay)`	Pause the program for 'delay' milliseconds.

Teacher Notes

- The program uses the potentiometer and 180 servo as part of the simulation. Remind students to use the divider board when connecting the potentiometer.
- Make sure the peripheral wires are connected correctly. Otherwise the peripherals will not work.
- Students should follow the instructions on the activity guide, and record their results.

Extensions

- For students with experience in programming, have them review the code for muscle_speed(). Explain how the code moves the servo back and forth. Make adjustments.

Cross-Curricular

- **PHYSICAL SCIENCE:** This program uses the 180 servo. Research the servo and how it works.

Mission 5: Muscle Magic
Objective 3: Neural Rhythms

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain how motor neurons create a rhythm.
- I can list factors that determine the speed of the rhythm.

Key Concepts

- Coordinated motor neurons take turns firing signals, which creates a rhythm.
- Action potential speed, influenced by neuron diameter, affects rhythm.

Assessment Opportunities

- Exit ticket: Give an example of an activity that requires coordinated motor neurons that would create a rhythm.
- Turn in the Activity Guide.
- Complete the program *neural_rhythms*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science: 9-12.AP.13 Create more generalized computational solutions using collections instead of repeatedly using simple variables.

Vocabulary

- **Rhythm generator circuit:** Neurons that take turns sending signals by inhibiting each other, creating a rhythm.

New Python Code

`from soundlib import *`	Access the audio functions from the soundlib library.
`drum = soundmaker.get_tone('noise')`	Set up an object to produce a sound. In this example, the sound is noise.
`drum.set_pitch(700)`	Set the tone, or pitch, of the sound object.

Teacher Notes

- The program uses the built-in CodeX speaker. If the room gets too noisy, you can suggest students use headphones. The CodeX also has a headphone jack just below the speaker.
- Students should follow the instructions on the activity guide, and record their results.

Extensions

- For students with experience in programming, have them review the code for and make predictions before running the code. Then see if they are correct.
- Students can make adjustments to the code:
  - Make the neuron rectangles bigger or smaller
  - Make the signal circle bigger or smaller, or a different color
  - Change the number of pixels the circle moves each time a signal is fired.
  - Adjust the variable dly as it is changed in drum_beat()

Cross-Curricular

- **PERFORMING ARTS:** Discuss or research rhythm in music or performance. Create different rhythms and discuss the mood each conveys. How could different rhythms affect the physical body or muscle movement?
- **VISUAL ARTS:** Create an art project that shows a rhythm.
- **SOCIAL STUDIES:** Research music in a specific region. What instruments do they use? How do they incorporate rhythm? Do research either historically, through time, or culturally, studying the impact of music in society.

Mission 5: Muscle Magic
Objective 4: Muscle Response

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain the theory that rhythmic movements could be generated by reflexes.
- I can explain how the CGP sets a rhythm in motion.

Key Concepts

- The Central Pattern Generator sets the rhythm of a motion.
- A reflex can interrupt a rhythm if a problem occurs.

Assessment Opportunities

- Exit ticket: Give an example of a motion that is set by the CPG.
- Turn in the Activity Guide.
- Complete the program *chewing_sim*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Physical Science: Science & Engineering Practices: Engaging in Argument from Evidence Evaluate evidence behind currently accepted explanations or solutions to determine the merits of arguments.
Crosscutting Concepts: Patterns Empirical evidence is needed to identify patterns.

Vocabulary

- **Central Pattern Generator (CPG):** The part of the motor neural network that sets the rhythm of a motion and keeps the rhythm going.

New Python Code

`display.fill(RED)`	Fill the LCD screen with a solid color

Teacher Notes

- The program uses the potentiometer and 180 servo. Remind students to use the divider board with the potentiometer. Also remind them to make sure the wires are connected correctly.
- Students should follow the instructions on the activity guide, and record their results.

Extensions

- For students with experience in programming, have them review the code for muscle_open() and muscle_cose(). How are they alike? How are they different? Try writing a single function instead of two.
- Students can make adjustments to the code, such as the timing, the angles of the servo, or the colors used.
- Change the program to simulate a different motion other than chewing. Adjust the text to describe the motion.

Cross-Curricular

- **LIFE SCIENCE:** Think about other actions that might use a CPG. What are things that could interrupt the rhythm, and what would fix the problem so the rhythm can resume.
- **LANGUAGE ARTS:** Explain how the Central Pattern Generator works, using your own words and explanation.

Mission 5: Muscle Magic
Objective 5: Athlete Reaction

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain how motor neurons are recruited by the brain to control muscles.

Key Concepts

- Tiny motor neurons are recruited for tiny muscles that do not tire easily, like posture muscles.
- Increased physical activity with larger muscles requires larger motor neurons. These larger muscles fatigue more quickly.

Assessment Opportunities

- Turn in the Activity Guide.
- Complete the program *athlete_reaction*.

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science: 9-12.DA.10 Create data visualizations to help others better understand real-world phenomena.

Physical Science: HS-PS4-5. Communicate technical information about how some technological devices use the principles of wave behavior and wave interactions with matter to transmit and capture information and energy.

Vocabulary

- None

New Python Code

`display.draw_line(0, 229, 239, 229, RED)`	Draw a line between point(0, 229) and point (239, 229)
`run_type, col = determine_type(velocity)`	The two values returned by the function are assigned to two variables in the order given.

Teacher Notes

- The program uses the potentiometer. Remind students to use the divider board with the potentiometer. Also remind them to make sure the wires are connected correctly.
- Students should follow the instructions on the activity guide, and record their results.

Extensions

- For students with experience in programming, have them review the code for get_speed(). Change the function so that the test starts when the knob is on the right and turns to the left. Record your times. Does it make a difference which way you turn?
- Change the program to simulate a different motion other than running. Adjust the text to describe the motion.

Cross-Curricular

- **MATH:** This program uses waves to represent data. There are many math applications that involve waves. Add a lesson about waves and graphing, or calculating the frequency, etc.

Mission 5: Muscle Magic
Objective 6: Reaction Revolution

Time Frame: 45 minutes
(Suggested time for CodeSpace Lesson)

Learning Targets

- I can explain the difference between a reflex and a reaction.
- I can explain ways to improve brain focus and performance.

Key Concepts

- Reflexes bypass the brain and are faster than reactions.
- Reaction time is the time it takes to respond to a stimulus voluntarily.

Assessment Opportunities

- Exit ticket: Discuss a way to improve reaction.
- Turn in the Activity Guide.
- Complete the program *train_your_brain.*

Success Criteria

- ☐ Complete the CodeTrek steps
- ☐ Program runs correctly without errors
- ☐ Activity Guide is completed

Standards

Computer Science:	9-12.AP.12 Design algorithms to solve computational problems using a combination of original and existing algorithms.
Physical Education:	PE-HS1.1.3 Students demonstrate knowledge of and competency in motor skills, movement patterns, and strategies needed to perform a variety of physical activities.

Vocabulary

- **Neuroplasticity:** The brain's ability to change and adapt in response to stimuli.

New Python Code

`n_pod = random.choice(pod_list)`	Select a random item from a list.
`msg = f"pod_light({n_pod}, {choice})"`	Using an f-string to format a string with variables.

Teacher Notes

- You can divide the class into teams, and each team can run their own trials at the same time. The program can run without pods. Just press BTN-A and/or BTN_R instead of a tap.
- If you divide the class into teams, put each team on its own non-adjacent GAME_CHANNEL.
- The time for medium and hard can also be adjusted, if the timing seems too fast or too slow.
- The activity guide has a chart for students to fill out with their trial statistics. Instead, students can create a spreadsheet and use it for data analysis.
- There are many test alternatives students can try, other than what is suggested on the activity guide:
  - Run tests with the dominant hand and then non-dominant hand. Compare the results.
  - Switch the "no-go" color so it isn't the same every time: one trial RED, and the next trial GREEN. How does this affect your focus?
  - Move the pods in different configurations for each test: clustered together, in a line, spread out in a circle, etc. Which configuration gives the best results? (see Fitt's Law)
  - Change the NUM_SEQUENCE for each trial, so that each trial is longer and longer. You only need to change the code on the Control CodeX, and not all the pods. Does the length of the trial and number of "no-go" colors (NUM_SQUENCE) make a difference?

Extensions

- Review the code for control_pod(). Near the bottom of the code, the hit_percent and score are calculated. Change the calculations. Find the error percent instead, and come up with your own scoring system.
- Change the code to get a random number for each delay.

Cross-Curricular

- **MATH:** Research Fitt's Law. The predictive model is usually used for a human-computer interface. Try experiments with Fitt's Law with a user interface, or adapt the model to the "train-your-brain" activity.
- **LIFE SCIENCE:** Research different reaction time tests with different stimuli, like audio or touch. Many examples can be found online, like this one. Then design and conduct more experiments.

Appendix A: Required Resources

Computer Resources

Each student will need:

- A computer with the Chrome web browser.
- Chromebooks work great – just make sure they are up to date.
- Windows 10 or Windows 11 will work with no additional drivers needed.
- A current Mac OS will also work with no additional drivers needed.
- A USB port is used to connect and program the CodeX. The CodeX comes with a USB to USB-C cable. If your laptop or computer has any other configuration, you will need a cable that has USB-C on one end.

Software Resources

- The interactive textbook and text editor is web-based. Make sure the website is not blocked.
- An email is required for signing in and saving work. It can be a gmail account, but any email will work.
- A per device license is needed to access the curriculum.

Physical Resources

The missions can be completed by individual students or student pairs utilizing pair programming. Each student or student pair needs:

- A CodeX device and USB-C cable
- A license for the curriculum in CodeSpace
- A set of peripherals from the Brain Kit:
    - A NeoPixel Ring and connecting wires
    - A potentiometer and divider board and connecting cables
    - A 180 servo and connecting cable
- Materials needed but not included:
    - Decks of cards or slips of paper for neuron signals (Mission 2, Objective 2)
    - A flashlight or light source (Mission 3, Objective 3)
    - (optional) 4 AA batteries for the CodeX

The classroom needs a set of at least 5 pods for the CodeX devices. The pods can be used as class, or shared between groups of students for the final activity (Mission 6 Objective 6). You can also 3D print more pods. The stl file is available in Thingaverse.

Notes

- When the CodeX is plugged into a computer, it will appear as a USB mass storage device, similar to a flash drive. This is not required for normal classroom use. So don't worry if your school has a policy preventing flash drives. You just close the pop-up window and continue.
- Occasionally Firia Labs will provide a software update that requires updating the core software on the CodeX. At those times you will need the flash drive feature to update the software, so you will need to use a computer with USB drive access. Often a teacher's computer is used to update all the CodeX.

Appendix B: Our Approach

Physical Computing and CodeSpace: a web-based professional-learning platform

Hardware brings code to life! Our versatile physical computing devices and peripherals get students excited about code. Our CodeSpace learning environment enables them to step up to computer science with real-world text-based Python coding. We include ready-to-teach standards-aligned curriculum with hands-on projects that motivate students.

While there are some great online coding educational programs, we think our approach helps reach a broader range of students. Our approach:

- Gets students focused "off-screen," programming with physical hardware that connects and interacts independently of their computers.
- Teaches a real, professional programming language. Even younger students appreciate that you can make real money with these exact skills. If they can read, and they can type, they can code in text-based Python.
- Gives students the tools to create *anything* they can imagine. Beyond projects and curriculum, we give students a full-fledged software development environment. These are professional-strength tools for writing code. Instead of a game-playing environment, students can "win with code" through engaging hands-on projects and their own creativity.

Project Based Motivation

Students may wonder why they are learning to code. We all find that knowledge tastes so much better when you're hungry for it! Our goal is to **motivate** students with tangible, challenging and practical **projects**...that just so happen to require coding to build. We want students to think about how they might code a given project using what they already know. Only then do we teach *just enough* coding concepts to help them get the job done. This approach gives reason and meaning to each concept, as well as relevant problem context, which helps them retain it.

Type it In

Students are often tempted to just copy and paste from lesson examples. Prior to our extensive testing of the curriculum on groups of 4th through 12th grade students, we were concerned that the typing burden might be a problem. But we were willing to risk it.

- Typing in the code forces focus, dramatically improving retention.
- Keyboarding proficiency is key to expressiveness in using a programming language.
- Mistakes in structure, grammar, punctuation, capitalization, etc. are priceless learning opportunities.

Students learn an incredible amount from their mistakes. Our goal is to provide awesome safety-nets for them, guiding them to iterate quickly through successive failed attempts to arrive at a working solution. Extensive classroom observation has convinced us that the typing burden is not a problem. Students dive right in, and they don't have to be speed typists to make great progress in coding.

Exploration and Creativity

One of the great things about coding is the expressiveness it affords. Coding is a craft that takes time to master, but with only a few basic tools you can start crafting some pretty amazing things! Before they even complete the first project, some of your students will probably be experimenting "off-script" with some ideas of their own. That's a good thing! Remember that students are learning programming skills they can use to build *any* application – from controlling a rocketship to choreographing dance moves. Nurture creativity, explore, and instill the joy of coding!

Appendix C: Teacher Resources

If you and your students are still fairly new to text-based coding, don't worry! Like other physical devices and their curriculum, we've designed the Brain Decoded with CodeX Kit and this curriculum guide to gently guide you from absolute beginner to a very comfortable level of proficiency. Remember this – Don't Panic 🙂

We understand that tackling a subject like Computer Coding can be pretty intimidating. Fear not, we've built some amazing tools to help you! As you begin this journey, know that the team at Firia Labs is here to help, too. If you run into any problems, just let us know and we'll get you back on track.

Classroom Preparation

Many objectives can be completed individually. There is also a place for pair programming and collaboration in the thinking classroom. Such practices foster knowledge sharing and give students a chance to communicate about what they are learning and reflect on their practices. It builds confidence and keeps students focused on the task. Pair programming can result in better quality work with less errors, and keeps teams "in the flow".

You may need to think about a balance between individual work and pair programming to give your students the best opportunities to succeed and truly engage in and enjoy this learning experience.

Daily Routine

Each lesson has a suggested time frame. Most Objectives, starting with Mission 2, are designed for one 45-minute class period. Some missions may go even longer, depending on the time you have to spend, extensions and cross-curricular activities, the abilities of your students, etc.

This mission pack has a lot of flexibility. For maximum comprehension, missions should be completed in order, but the amount of time spent on each mission is up to you.

Extensions and Cross Curricular Activities

Each lesson has extension and cross-curricular activity suggestions. You can choose if you want to include any of them with your class. You may even have your own ideas! If so, plan on more time for the lesson.

Managing a Class

Our CodeSpace learning platform makes it easy for you to create a class for your students to join, and enables you to monitor their progress.

For help and step-by-step instructions, visit: https://learn.firialabs.com/curricula/code-space

If you are a **Google Classroom** teacher, you can import assignments from CodeSpace into your classes. For instructions, go to "Virtual Tools with CodeSpace" in the Teacher Resources for Python with Robots.

If you need assistance for anything, please send an email to: support@firialabs.com

Here are the basics of the CodeSpace Teacher Dashboard

- Log in to CodeSpace and from HELP, select CLASS DASHBOARD
- Once you are in the dashboard, click + in the green bar, top right corner, to add a class.
- Assign each class a name, and allow members to join with a join code.
- You can assign Google Classroom as your LMS.
- After the class is created, you can edit the class, get a join code, disable joining, etc.
- You can delete a student using the “remove” function.
- Students go to CodeSpace and click the SELECT CLASS button.
- They can click the JOIN CLASS button and enter their join code for your class.
- The class will be activated and they are ready to start working!
- In the dashboard, you can see student progress, as a whole class and individually.

Class dashboard

Individual progress