Lesson #3:

Computer Science with the Shake

Lesson Objective:

Use Raspberry Shake to understand core Computer Science principles and lay the groundwork for future coding projects with the Shake.

Learning Context/Anticipatory Set:

Before instruction, spark the students interest with having a “Shakemeter” running live on your screen (follow the tutorial here to build it). Tell the students that this is the next exciting direction to go in with the Shake – computer science and programming!

Key Terms:

- Computer: A device that accepts data as input, processes that data using logical processes called programs, and outputs the processed data
- Input: When a computer or device is receiving a command or signal from outer sources
- Output: The signal that the computer sends out to another device or system, or to the screen
- Process: The instance of a computer program that is being executed
- Analog: Of or pertaining to continuous variations or transmissions of a continuous signal
- Digital: Of or pertaining to discrete measurements or approximations that can be stored by a computer
- Network: A set of computers connected together for the purpose of sharing resources
- Internet: The global information system that is logically linked together by a globally unique address space, based on the Internet Protocol (IP)
- UDP (User Datagram Protocol): A protocol for sending data over a network

Direct Instruction:

Basics of Computer Science – With Raspberry Shake

Use Raspberry Shake to understand core Computer Science principles and lay the groundwork for future coding projects with the Shake.

To fully understand the Raspberry Shake, you must understand at least the very basics of computer science. In this lesson we will go through those basics, in order to set the foundations for opportunities in the future to do community projects. So, let's start with the basic question:

What is a computer? Watch the video below to learn more -

> **Code.org made an excellent video explanation of what makes a computer a computer. Watch and share with your students! Watch the video at https://youtu.be/xfKn5OjHLqQ**

In summary, computers receive an input, process and store information, then provide an output.

It is important to understand that all data and information that computers can deal with is *digital* data. It is expressed, at least in the computer's inner workings, as ones and zeros, and decoded and encoded to make it easier for humans to read. The real-world complement to digital data is analog data, which is just a constant stream of information represented by a continuously variable physical quantity. Digital data can approximate analog data using a process known as sampling, which you can think of as putting many points along a curve and connecting those points like connect-the-dots. The analog signal and the digital signal will never perfectly sync up, but the digital signal is a good approximation.

We can understand these concepts very well through understanding the Raspberry Shake, and the components that make it up. We have the Raspberry Pi, which is the main "computer". We also have the geophone sensor and "Shake Board" digitizer as part of the Raspberry Shake device. The geophone responds to *analog* data signals, because of its ability to turn ground movement into a constant stream of voltage information. Then, that voltage is sampled by the digitizer, which means it records a data point by converting the data from voltage (analog) to digital. This process of quantifying voltage data is the heart of the process of recording seismic data.

The class will take these concepts and use them in creating a coded program to process and display the Raspberry Shake's UDP data!

Practice:

Time: Rest of class (and next class) – Start with Node-Red connectivity

Students get out a computer (laptop, Raspberry Pi, etc.) to start Node-Red Server on. Students can work together in pairs.

Follow this tutorial (https://edu.raspberryshake.org/classroom-curriculum/node-red) to get started.

**If there is less than 20 minutes remaining in the class, go ahead to closing and complete the activity next class.

Closing:

Time: 5-10 minutes — Written Reflection

Students write a short, 3-5 sentence reflection on what they learned and any insights they made.

- Did today's lesson change your perspective on computer science?
- Did it increase your curiosity?
- Was it difficult to understand?