Sample Curriculum Unit

The following Sample Curriculum Unit is an example that may be used in a classroom for one 90 minute or two 45 minute sessions

PancakeBot – Introduction to Computer Programming

Primary Subject Area and Grade Level: Math, grades 3 - 5
Interdisciplinary Connections: Technology, art
Lesson Duration: One 90 minute session
Relevance/Rationale: In today's economy, 21st century skills that have STEM or STEAM applications are more important than ever. Many children in schools, library programs, or other enrichment classes do not get opportunities to learn programming languages or write code. They do not know that they, too, can learn how to code and become a part of 21st century innovation. For the average American child, a lifetime interest in computer programming and other STEM-related work begins with experiences like PancakeBot.

Outcomes/Objectives:
Use G-Code to learn how to control PancakeBot.
Use the coordinate plane to create horizontal or vertical lines in G-code.

Content Standard(s) and/or Common Core Learning Standard(s):
CCSS.MATH.CONTENT.3.MD.C.5

Recognize area as an attribute of plane figures and understand concepts of area measurement.

CCSS.MATH.CONTENT.5.G.B.4

Graph points on the coordinate plane to solve real-world and mathematical problems.

Use of Formative Assessment to Inform Planning:

After demonstrating competency with using paper and pencil to plot and connect points on the coordinate plane, students will apply the skill to writing in G-code to create lines with PancakeBot.

Class Information: Will vary. Unique student populations may include students with special needs, ELL students, and gifted students. The PancakeBot addresses multiple learning styles, and will appeal specifically to the kinesthetic learner. Visuals that connect the programming commands to what is printed on PancakeBot will be key in reaching all students.

Overview:

In this lesson, students will relate their earlier work with the coordinate plane to the PancakeBot. The teacher will demonstrate, using this exciting technology, a real-world application of this math skill. The teacher will show students how to use G-code to communicate with PancakeBot to print out horizontal or vertical lines on the machine's coordinate plane.

After understanding the basic commands needed for printing out a horizontal or vertical line on PancakeBot, students will work in pairs to write the code. The teacher and/or other adults will circulate among pairs to give real time, constructive feedback on whether their code will work.

Students can then assess each other's code, making any changes as needed. Once they are happy with their work, they can use PancakeBot one pair at a time to enter their code.

When all students have had a chance to enter their code, they can approach the PancakeBot one pair at a time to test out their code. Teacher or other adults in the room will assist with turning the machine on and off, filling the machine with pancake batter, clean up, etc. Each pair of students will create two lines using their code, so that they can eat the results afterwards.

At the end of the lesson, students will complete a short assessment, the Sample Coding Scenario sheet, that presents them with the following question: *If Student X wrote the following code for PancakeBot, would it produce a horizontal line? Why or why not?*

Technologies and Other Materials /Resources:

Teacher: Laptop, LCD projector, big G-code chart, PancakeBot, premixed batter, towels, forks, plates

Students: 1 laptop per pair, G-code sheet, Sample Coding Scenario sheet, plates, forks, assessments

Grouping Strategy: Students can be paired according to their comfort level with the coordinate plane, their comfort level with technology, or their comfort level with programming languages. Goal-oriented students with greater stamina or a greater ability to try new challenges can be paired with students that need more support and encouragement. If all students are close in ability, pairing them according to stamina and readiness for challenge may be a good strategy.

Academic Vocabulary:

Programming language
G-Code
Zeroing
Extrusion
Polygon
Pressure
Vacuum Pressure
Calibration
Feed Rate
Standard
CNC

Lesson Procedures:

<u>Lesson Launch</u>

Begin the lesson by asking students about their favorite video games, songs, TV, shows, etc, any media that is technology related. Then ask them if they've ever wondered what it takes to make those things work, or if they believe there are people behind the scenes that keep things running.

Explain that today, they will learn about a programming language called G-code that makes the PancakeBot work. Let students know that learning a programming language is similar to how they learn other languages. It takes time to memorize the main words to use, and it takes time to be able to use it fluently! Ask for a couple of students to comment on what it's like to learn and use a new language. Remind them that we will all be writing in and speaking in a language that is new to us today. We will be

unpacking the secret of how to talk to computers to get them to do what we want them to do.

Background Knowledge/Orienting Students

Show students what G-code looks like on a computer screen. Describe the history of G-code and explain how the code commands the PancakeBot to make different shapes and pictures:

A brief history of G-code

G-code was first implemented at MIT Servomechanisms Laboratory in the late 1950s[1]. It was standardized in the late 1960s and is recognized around the world. G-code is used primarily for CNC machines which sometimes hide the G-code but PancakeBot does not hide the G-code leaving it open for you as an artist, engineer, maker or inventor (amongst other things), to write your own G-code for PancakeBot.

How G-code works

A flow chart or PowerPoint will provide the visual support for explaining how G-code works with PancakeBot:

PancakeBot receives commands from G-code. G-code is a language in which people tell computerized machine tools how to make something.[2] In our case, G-code tells PancakeBot when to extrude, or release, batter, where to extrude it and how long to pause between extrusions amongst other things. Other commands allow for 'zeroing' of the PancakeBot head and feed rate. The G-code commands are stored in a text file that are read from the SD memory card and tell PancakeBot when to extrude or not extrude batter and where to move over the griddle. The G-code commands can easily be read by opening the .gcode file created by Pancake Painter or the G-Code file can be created by hand using simple G-Code commands.

G-code and the Coordinate Plane

Continue with flow chart or PowerPoint to provide the visual support for explaining how G-code can be used to plot and connect points on the coordinate plane:

Explain that G-code helps us plot points and connect them on the PancakeBot's griddle, which can be imagined as a coordinate plane:

[1] https://en.wikipedia.org/wiki/G-code
[2] https://en.wikipedia.org/wiki/G-code

PancakeBot has an X and a Y axis and moves the bottle to different locations by using Cartesian Coordinates. The maximum horizontal X coordinate is 440 and the maximum Y coordinate is 240. This is measured in millimeters so your print area is 440mm by 240mm. That's a pretty big pancake! Coordinates are arranged in standard form of (X,Y) where the X coordinate is written first and the Y coordinate is written second. Pancakes are drawn on the top right quadrant also known as Quadrant I or the First Quadrant.

Have the G-code commands posted on a chart that is visible to students. Explain that the chart will be referenced by everyone, because the language is new for everyone today.

Modeling

Use G-code to draw a simple horizontal line. Refer to the G-code chart as you enter commands on the laptop, explaining very explicitly why you are choosing to enter each command and what it will do:

A simple G-Code example for drawing a horizontal line from coordinates (20,10) to (220,10) would be as follows:

G28; Home all axis. This lets PancakeBot start at the same spot all the time.
G00 X20 Y10; Moves PancakeBot head to coordinates (20,10)
M106; Turns the extruder on
G4 P750; Pause PancakeBot for 750 milliseconds or three quarters of a second. This pause is necessary to allow the batter to exit the bottle and hit the griddle
G00 X220 Y10; Moves PancakeBot head to Coordinates (220,10)
M107; Turns extruder off
G4 P250; Pause PancakeBot for 250 milliseconds or one quarter of a second. This pause is necessary to allow the batter drop to separate from the bottle before moving.
G28; Home all axis. Tells PancakeBot head to return to corner.
M84; Releases all motors (except for vacuum pump) allowing user to move PancakeBot head by hand.

Checking for Understanding

- Why do we begin and end with G28?
- Why does G00 have to be used twice to draw a line?
- How are the points on the coordinate plane identified in G-code?
- What would happen if we forgot to include the G4 commands?

Guided Practice

Ask the class to help you draw a vertical line. Ask them what commands should be entered first, second, etc., filling in what they are not clear on.

Independent Practice

Students will pair up, writing code on their laptops to tell PancakeBot to draw either a horizontal or a vertical line. Teacher will circulate to answer questions and give students feedback on any commands they may be missing.

As pairs finish, the teacher will help them enter their code into the PancakeBot. Once all pairs have entered their code into PancakeBot, students can test out their code one pair at a time. Give students their horizontal or vertical lines to eat.

Reflection & Wrapping Up

What was the most challenging aspect of writing code for PancakeBot? What was fun or exciting? How long do you think it would take to become comfortable writing in G-code?

Have students complete the Sample Coding Scenario worksheet.

Differentiation:

Students with special needs can work with the teacher or another adult in the room as needed to go over the instructions for the assignment and do the steps together, one-on-one. ELLs will benefit from being shown a variety of graphics during the modeling portion of the lesson. High achieving students can challenge themselves by creating beyond horizontal and vertical lines.

Assessment Criteria for Success:

After the modeling, teacher will assess student understanding. During the independent practice, teacher will visit each pair of students, refining their code as needed. Students will show success with lesson outcomes when their pancakes have been printed to their specifications. Formal assessment at the end of the lesson will assess understanding by asking students to apply their skills to a sample coding situation.

Anticipated Difficulties:

Many students will be new to computer programming. Start the lesson by appealing to students' sense of how the world works. They will quickly see that their favorite songs, shows, art supplies, sports equipment, video games, etc. all require technology to exist. Using lots of visuals, step-by-step instructions, and strong teacher modeling to give students confidence with programming will help.

Some students will not work well in pairs. Constant circulation of the teacher and any other adults in the room will help to keep groups focused and productive.

Stamina may also be an issue. If students are struggling, provide more support by talking them through the process step by step.

Reflections: List at least three questions you will ask yourself **after** the lesson is taught.

Did the students enjoy writing code for the PancakeBot?

Did the students make the connection between the math and the technology? Did students' attitudes towards technology change as a result of this lesson?