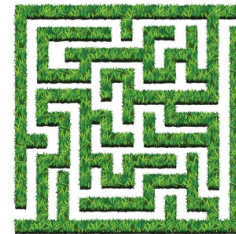


Program a Maze with Paper



Inspired by Program a Maze with Paper from Science Buddies

STEELS Standards

- [3.5.6-8.E](#)
- [3.5.6-8.Q](#)
- [3.5.6-8.T](#)

Objectives

- Introduce the basics of programming (bugs, commands, relationship between person and program)
- Hands-on learning: by physically manipulating paper mazes and coding sequences of commands to navigate them, participants engage in experiential learning, which can help their understanding of programming concepts
- Problem-solving skills: Participants are encouraged to think critically and strategically to solve the maze challenges presented in the activity

Materials

- [insert mazes here](#)
- Ruler (if you are creating your own maze)
- Pencil

Basic Vocab

- **Program**
 - At its most basic level, a program is a set of instructions that tells a computer what to do.
- **Bugs**
 - Errors in computer code that lead to unexpected behavior
 - Similar to how you may make a mistake when following a recipe, computer programs can misinterpret the directions and act differently than how they're supposed to
- **Debugging**
 - The process of finding and removing bugs.

Introduction

Begin by asking students what they think a computer program is.

Then explain what it actually is:

- You probably use computer programs every day. Every time you play a video game, or use a phone or computer, you are using computer programs. These programs are a set of instructions that tell the computer what to do. For example, if you click the "like button" on an app, there's a program that tells your device to increase the like count or turn the heart from white to red. There are many examples of what computer programs can do. Can you think of more?
- Computers are very good at following directions. Programmers are responsible for making these directions and telling computers what to do. But what happens if a programmer gives the computer wrong instructions by accident? Computers don't have a brain like you and I do, so they will follow these directions even if they're wrong. This can cause a program to "crash": behave incorrectly or sometimes stop completely. Programmers call these errors "bugs" and "debugging" is when you find and remove these problems.

Introduce the activity:

- In this activity, you'll learn more about creating a program to solve a maze. In other words, you will be creating a set of directions to help a computer find their way through a maze without crashing or getting lost. If you make a mistake, you will "crash" into the wall of the maze or get lost and you'll have to debug your program and try again!

Class Activity

Preparation:

1. If you are creating your own maze, use a piece of graph paper and a ruler to create a small maze within a square/rectangular grid.
 - a. All of the lines in the grid should be either vertical or horizontal (no diagonal or curved lines).
 - b. Clearly label a "start" and "finish" point.
2. If you are using a premade maze, print them out.
3. Start with smaller, easier mazes and then you can create more difficult ones later.

Paper Maze Instructions:

1. Imagine you are a person navigating through a maze starting at the "start" point. On a piece of paper, write down step by step directions how you would move through the maze. For example, "move forward 2 squares" or "turn left".
2. Continue writing directions until you can reach the finish point.
3. Test your maze! Follow the directions EXACTLY how they're written, even if they are incorrect to simulate how computers follow directions.
 - a. One way students can do this is by swapping their list of directions with a partner and having each student follow the other's directions and see if they successfully navigated the maze.
 - b. Another option would be to ask a student to volunteer and you (the instructor) can use their directions to try to navigate through the maze in front of the class.
4. If you could successfully make it to the finish line, congratulations! However, if your directions led you to crash into a wall, your program has bugs and you need to debug your program.
5. To debug your program, carefully go through your program step by step to see what went wrong. Even one small mistake can mess up your whole program, so look very closely!
6. If you did have bugs, rewrite your program and try again.

maze.

b. Another option would be to ask a student to volunteer and you (the instructor) can use their directions to try to navigate through the maze in front of the class.

4. If you could successfully make it to the finish line, congratulations! However, if your directions led you to crash into a wall, your program has bugs and you need to debug your program.

5. To debug your program, carefully go through your program step by step to see what went wrong. Even one small mistake can mess up your whole program, so look very closely!

6. If you did have bugs, rewrite your program and try again.

Discussion

(Try to guide student discussion to touch on these)

- Discuss the importance of telling the computer exactly what to do and what happens when there are bugs
 - Bugs can cause a program to crash
- Discuss more generally how computers differ from humans
 - We can usually use reason to figure out what to do, but computers/programs can't think on their own. That's why they need a human to give clear directions about what to do.
 - Even in instances of AI, such as ChatGPT, a real person codes and programs it
- Discuss how computers have changed the way people think, interact, live, and communicate
 - People need to be good problem solvers in order to work with computers
 - Just like how in the maze activity, it can sometimes be difficult to strategize and plan the route through the maze, while anticipating potential obstacles and making decisions to navigate effectively
 - The use of computers has encouraged people to think more creatively. Often times in programming, you have to experiment with different strategies and approaches.

Additional Resources

[Maze Game on Code.org](#) (highly recommended)

- Students can use the fundamental ideas from this activity and apply it to this game on their devices.
- The maze is a bit more complicated and set up as a game instead of just a maze.
- It utilizes code blocks and tests the student's code to see if they completed the levels successfully.
- This game just helps further connect the paper maze activity to actually programming but uses the same concepts.