



Long Activity
Ages 6–8
30–45 minutes

Coding a Maze

Description

Learn the basics of coding language as you guide a small figurine through a paper maze, and then try out your new coding skills on the computer

Space Considerations

Participants will need access to tables and enough space to spread out the coding printouts. If the computer coding portion of this activity is included, all participants will also need access to a computer or tablet (individually or shared in small groups).

Competencies

- Computer literacy
- Logic and sequencing
- Spatial relationships
- Strategic thinking

Materials

- Paper for printing maze templates
- Paper for printing coding commands
- Characters (figurines similar in size to a LEGO person)
- LEGO or DUPLO (optional)
- Computers or tablets (optional)

Preparation

- Print several copies of the coding commands—about one copy for every two participants
- Separate the coding commands by cutting along the grey lines
- Print out maze templates, one for each participant

Implementation

1. Optional: Participants can add LEGO or DUPLO to the grey parts of the printed mazes
2. Hand out Maze 1
 - Introduce the concept of "sequence":
 - Code must be written in a specific order called a sequence
 - Just like a story wouldn't make sense if the sentences were re-arranged in the wrong order, code won't work if it's written in the wrong sequence
 - This applies to the maze and also to coding more generally
 - For this maze, hand out the following pre-cut coding instructions: *Go Forward, Turn Right, Turn Left, End*
 - Participants need to create a long list of the instructions that they think the character needs to follow to reach the end of the maze
 - The character must avoid the grey areas of the maze and only stay on the white squares
 - Once participants have organized their list from top (first command) to bottom (final command), place the character at the start of the maze
 - Another participant, staff or a caregiver can check the code by moving the character and following the instructions in the list
 - Flip over each command as it is completed
 - Participants can make corrections as needed—young children often need corrections on left and right turns, as they must think about the turns from the characters' perspective
3. Hand out Maze 2
 - Introduce the concept of "loops":
 - This is when you want to repeat steps in a sequence
 - Rather than piecing together three separate "move forward" commands, children can learn to use the code "for the next _ steps, move forward," filling in the blank space with the amount of steps needed
 - Hand out the following pre-cut coding instructions: *Go Forward, Turn Right, Turn Left, For ____ Steps, End*
 - Participants need to create a long list of the instructions that they think the character needs to follow to reach the end of the maze
 - It's a good habit to start indenting the line of code underneath the loops—this is required by some computer coding languages, and it also makes the language much more readable
 - Once participants have organized their list from top (first command) to bottom (final command), place the character at the start of the maze

- Another participant, staff or a caregiver can check the code by moving the character and following the instructions in the list
 - Flip over each command as it is completed
 - Participants can make corrections as needed
4. Hand out Maze 3 (Note: a more advanced concept; for older children)
- Introduce the concept of “if-then-else” statements:
 - These will enable kids to think about writing as short a program as possible
 - An if-then-else statement is comparable to answering a *true or false* question—if the answer is true, a certain action occurs; if the answer is false, another action occurs
 - To get their character to walk in a straight line, participants could come up with the following code: “If > there is no wall > in front of me > go forward”
 - Hand out all of the coding instructions: *Go Forward, Turn Right, Turn Left, For ___ Steps, If, Else, Else If, There is a Wall, There is Not a Wall, In Front of Me, To My Left, To My Right, On All Three Sides*
 - Participants need to create a long list of the instructions that they think the character needs to follow to reach the end of the maze
 - There are many possible codes that can be written based on these options
 - Allow participants to experiment with the different options and see if they can write a variety of code to get their character to the end of the maze
 - There are many ways to get the character from start to finish, so keep on experimenting with different codes
 - Once participants have organized their list from top (first command) to bottom (final command), place the character at the start of the maze
 - Another participant, staff or a caregiver can check the code by moving the character and following the instructions in the list
 - Flip over each command as it is completed
 - Participants can make corrections as needed
5. If you have access to computers or tablets for the program, or if you would simply like to encourage participants to practice their coding after the program, try out the ten challenges on [maze blocky](#), which is a comparable exercise to the printed maze the kids have just completed

Book Suggestions

Awesome Minds: Video Game Creators by Alejandro Arbona

How to Code a Sandcastle by Josh Funk

I'm a JavaScript Games Maker: The Basics by Max Wainwright

Images and Templates

Coding Commands

Go Forward ↑	Go Forward ↑
Go Forward ↑	Go Forward ↑
Go Forward ↑	Go Forward ↑
Go Forward ↑	Go Forward ↑
Go Forward ↑	Go Forward ↑
Turn Left ↶	Turn Right ↷
Turn Left ↶	Turn Right ↷
Turn Left ↶	Turn Right ↷
Turn Left ↶	Turn Right ↷
Turn Left ↶	Turn Right ↷

FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END
FOR	 _____ steps	END

IF	there is a wall	in front of me
IF	there is a wall	in front of me
IF	there is a wall	to my left
IF	there is a wall	to my left
ELSE IF	there is not a wall	to my right
ELSE IF	there is not a wall	to my right
ELSE IF	there is not a wall	on all 3 sides
ELSE IF	there is not a wall	on all 3 sides
ELSE	END	END
ELSE	END	END

Start							
			End				

Start							
							End