

Identifying Misplaced Shelf Items with an LLM Bot and Camera

Ozobot Editor Lesson—Teacher's Guide

Grades 9-12

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

If you haven't already done so, you should read the five-page student guide that accompanies this lesson. It provides all of the information needed by you and the student to thoroughly understand the lesson, including the Ozobot Editor block-based program.

Lesson Summary

The lesson involves analyzing three shelves in a sporting goods store with potential misplacements. Using an LLM prompt, the system identifies the location of misplaced items and then guides an Ozobot through a mapped layout to highlight correct and incorrect placements with colored lights. The process includes parsing the LLM's response to locate misplaced items and programming Ozobot's movements accordingly. The goal is to develop a program that accurately detects misplaced objects and visually flags them, applicable across different shelf maps (X, Y, Z) without modifications to the core prompt.

Computer Science Programming Concepts in This Lesson

- The use of variables to hold the values of program constants
- The use of loops to handle tasks that must be repeated several times
- The use of conditionals, as needed, to control the flow of logic
- Variable names that support self-documentation
- Extracting information from a string by parsing
- The use of LLM (Large Language Models) with a camera

Share Code – 89v4tmj

Example Video

Accompanying this lesson is a video entitled *Misplaced Shelf Items.m4v*. Your students will want to view this video to gain a quick view of the ultimate result of their work on the lesson. It shows Ari moving on Ozomap X, displaying a red top light for misplaced shelf items and a green top light for items that are where they belong.

Applicable CSTA (Computer Science Teachers Association) Standards

2-CS-02	Design projects that combine hardware and software components to collect and exchange data.
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.
2-AP-19	Document programs in order to make them easier to follow, test, and debug.
3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3B-AP-08	Describe how artificial intelligence drives many software and physical systems