

OzoBlockly Editor—**Ozobot Identifies Playing Cards**—Teacher’s Guide

Grades 9 – 12

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

In this lesson, students will utilize the Ozobot and its camera to program Ari (or Evo) to identify playing cards through pattern recognition and multimodal AI. The lesson emphasizes hands-on experience as students set Ari to navigate a specific path, stopping at intersections to identify the suit of a visible card. Upon identification, Ari will beep and display the card name on its screen. Key elements include initialization, card identification, setting light colors according to suits, and executing functions for movement, beeping, and displaying messages. The program development encourages modular programming practices, allowing students to understand and manage complex tasks systematically. The lesson is augmented by a video demonstration that illustrates the process of card identification and offers insights into preparing for successful execution. Ultimately, students will gain practical skills in programming with Ozobot and a deeper understanding of AI capabilities in recognizing visual inputs.

If you haven’t already done so, you should read the *Ozobot Identifies Playing Cards Student Guide*. Doing so will familiarize you with the program details.

Programming Concepts Involved with Coding the Program

- The use of variables that support self-documentation
- 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
- The use of functions to support modular program development
- The use of an LLM (Large Language Model) with a Camera

Ways to Use This Lesson

1. Turn the students loose on the lesson, having them carefully study the Student Guide, enter the program in the Ozobot Editor, and testing the program on several different playing cards.
2. If time is a limiting issue, have the students carefully study the Student Guide but provide them with the **share code 73b3bhy**.
3. Show the students the Ozomap for this lesson, have them view the accompanying video, and let them develop a program completely on their own. This should be quite challenging for most students. This would best be accomplished as a homework assignment, perhaps giving them several days to a week to complete it.

4. Discuss the function that identifies the playing cards and how to develop an LLM prompt that simplifies the task of preparing the program. Then let the students develop the remainder of the program on their own. Also, encourage the use of functions that modularize the program. This should be a bit less challenging than option 3.

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-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
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.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
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.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.