

OzoBlockly Editor—**Lesson: Ozobot Identifies Playing Cards**—Student Guide

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

In this lesson, you will make use of Ari (or Evo) and an LLM Bot with camera. You will program Ari to navigate to one of four intersections on a larger version of the Ozomap shown in Figure 1. Ari will stop at the intersection representing the suit of the visible card, beep twice, and then display the name of the visible card within that suit on Ari's screen: Ace, 2, 3, ...9, 10, Jack, Queen, or King.

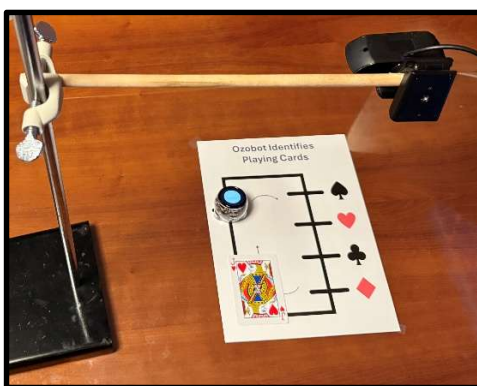


Figure 1

The camera is suspended from a ring stand as shown, so that it can see the card clearly. Of course, you can just hold the camera and view the image in the Ozobot Editor to make sure that the card is clearly seen.

A video that accompanies this lesson shows Ari identifying five different playing cards in sequence. **Ari should initially be placed on the far-left side of the Ozomap path near the straight arrow and facing upward.** Each time Ari reaches the intersection in the upper-left, you have 3 seconds to place the card that is to be identified on the lower left corner of the Ozomap. You should view this video now to get a good feel for how your program will work.

How is Ari Able to Identify the Playing Card?

The Ozobot Editor provides you with a *multimodal* model. This means that it can process images along with text. Through pattern recognition, the model scans the image for specific features including:

- Shapes (diamonds, spades, hearts, and clubs)
- Colors (red, black)
- Alphanumeric characters (A, 2, 3, ... 10, J, Q, and K)

The LLM models used by the Ozobot Editor have been trained on extremely large datasets of labeled images. As a result, they have likely seen numerous images of playing cards from a variety of angles, under different lighting conditions, and with many different designs. Suppose that AI identifies a specific combination of features, such as a diamond and a Q symbol. It would then map that to the playing card “Queen of Hearts”.

The Main Program and the Initialization Function

Having viewed the accompanying video, it should be clear that the program must do the following things for each playing card that is to be identified:

1. Identify the card with AI, using an LLM bot and a camera.
2. Set the top light of Ari to the color of the identified suit.
3. Move Ari to the correct suit intersection.
4. Beep twice at the intersection.
5. Display the card within the suit on Ari’s screen.
6. Return to the home position, so that the next card can be identified.

These six tasks are best accomplished through functions, making program development modular and much easier to follow. With this in mind, Figure 2 shows the main program.

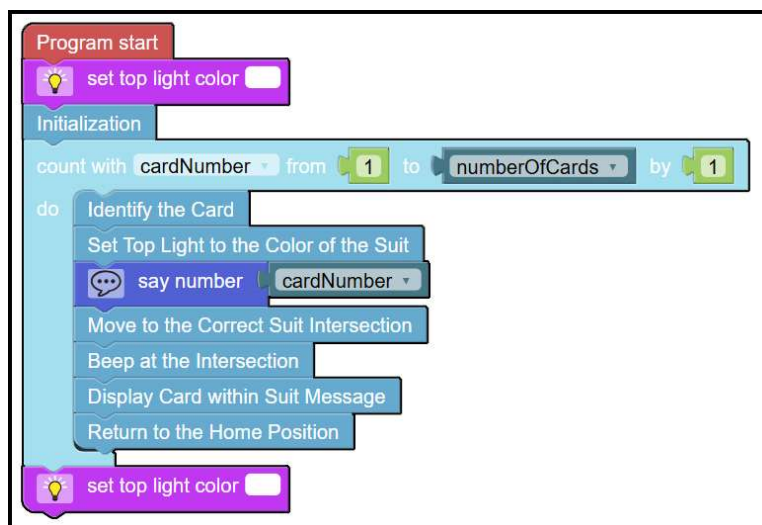


Figure 2

We start and end the program with Ari’s top light set to the color white. Recalling that Ari must initially be placed on the far-left side of the Ozomap path near the straight arrow and facing upward, the “Initialization” function accomplished this, moving Ari to the intersection in the upper-left. See Figure 3. We also initialize the number of cards to be identified to the value 5, or whatever you choose. As shown in the main program, a *count with* loop then takes us through identification of that number of cards. The card number is spoken before Ari moves to the correct suit intersection. This helps keep track of where we are in the loop.

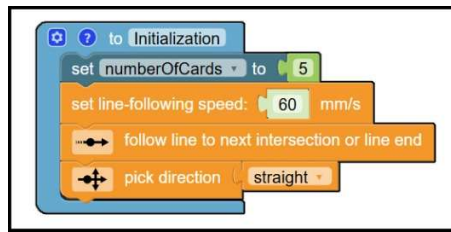


Figure 3

The “Identify the Card” Function

The function that identifies the playing card is where AI comes into play. This function is shown in Figure 4.

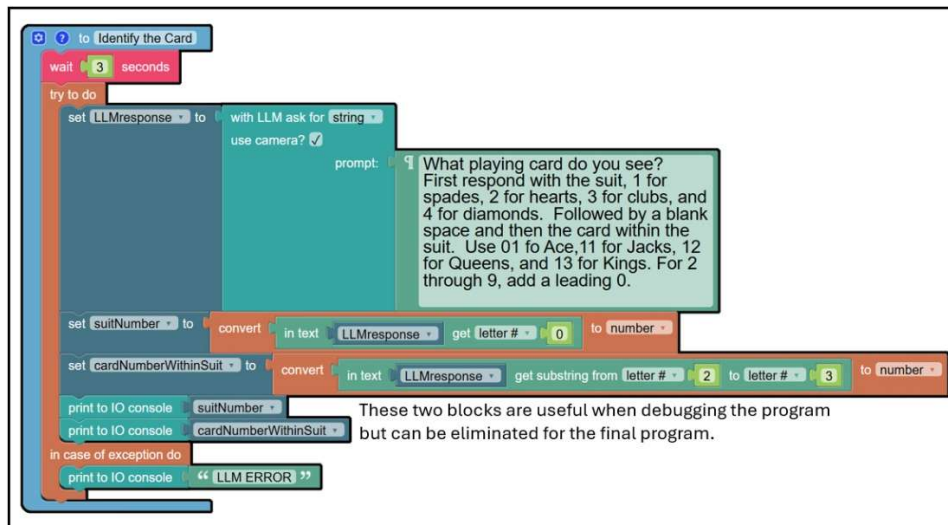


Figure 4

This function begins by allowing 3 seconds for the user to place the card in the lower-left corner of the Ozomap. The response from the LLM is set to be a string.

Suppose the card to be identified is the Queen of Hearts. If the prompt had been simply “What playing card do you see?”, then AI might return with a variety of responses, including, but not limited to “I see the Queen of Hearts”, “The card shown is the Queen of Hearts”, or simply “Queen of Hearts”. To make programming the task easier, we need to specify how the response should be constructed, so we add the rest of the prompt shown in Figure 4. The result would then be the string “2 12”. The 2 tells us the card is from the Hearts suit, and the 12 tells us that the card is a Queen. We then parse the string and convert the first character of the string (whose index is 0) to a number and get the suit number (2). The substring consisting of character indices 2 and 3 is then converted to a number to get the number of the card within the suit (12). With these two numbers in the variables *suitNumber* and *cardNumberWithinSuit*, the task of programming the remaining functions is much easier. Note that we asked the LLM to insert leading 0’s for Ace through 9 so that the card within the suit would always be two characters (01, 02, 03, ... 09) long as a string.

The “Set Top Light to the Color of the Suit” Function

Just as an added attraction after identifying the card, we will set Ari’s top light to the color of the suit, either red or black. The black suits, Spades and Clubs, have odd (1 or 3) values for the variable *suitNumber*, while red suits, Hearts and Diamonds, have even (2 or 4) *suitNumber* values. Pure black, in which (R,G,B) = (0,0,0) will always show white text, which we don’t want. So, we set (RGB) = (50,50,50), which shows up as a dark gray on Ari’s screen. This will have to do for black!

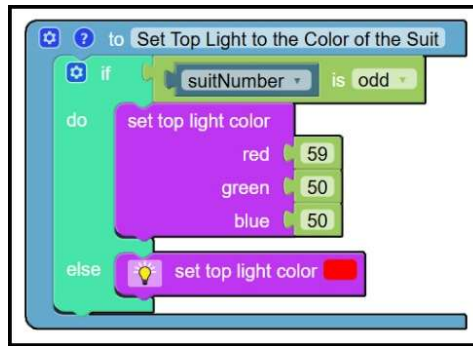


Figure 5

The “Move to the Correct Suit Intersection” Function

Figure 6 shows that this function is quite straightforward. To reach the correct suit on the right side of the Ozomap, either Spades, Clubs, Hearts, or Diamonds, Ari simply needs to advance the number of intersections equal to the *suitNumber*.

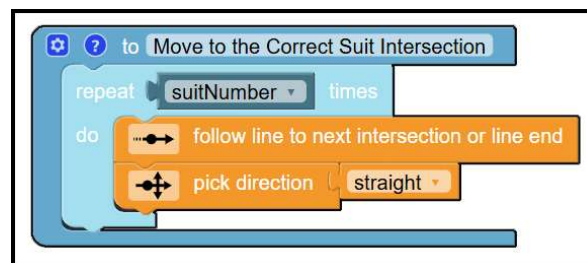


Figure 6

The “Beep at the Intersection” Function

This function, shown in Figure 7, serves the purpose of notifying the user that Ari has reached the intersection that represent the suit of the identified card.

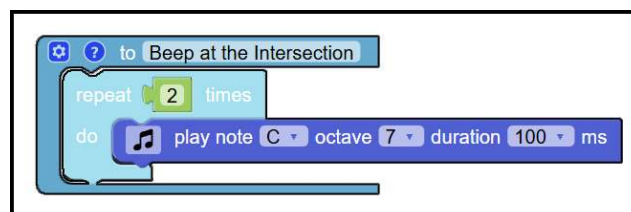


Figure 7

The “Display Card within Suit Message” Function

This function is shown in Figure 8. We make use of the logic **if** statement with three **else if**'s followed by an **else**. To improve program execution efficiency, the originating **if** first checks if the card number is between 2 and 10, inclusive, since this range is more likely than the card being an Ace, Jack, Queen, or King. The User IO “print message” block will display the card “number” (Ace, 2, 3, ..., 10, Jack, Queen, or King) on Ari’s screen or on the Ozobot Editor Terminal for Evo.

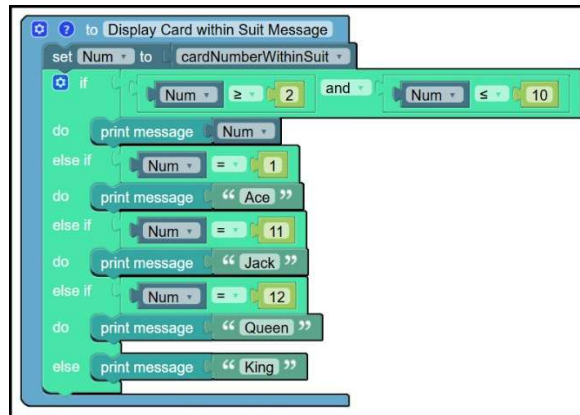


Figure 8

The “Return to the Home Position” Function

Figure 9 shows this function. The number of intersections to follow is inversely proportional to the suit number. If the suit number is 1 then Ari must travel 4 intersections before reaching the home position. If the suit number is 2 then Ari must travel 3 intersections before reaching the home position, and so on.

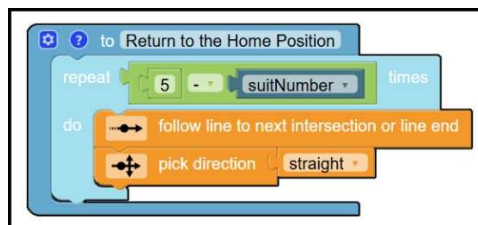


Figure 9

Working with the Program

Great job! You've successfully gone through the whole Ozobot Editor Blockly program, and all its subroutines (those special functions!). Now for the exciting part—let's get this program entered into the Blockly editor. After you've done that, it's time to test it out! Check if your program accurately identifies the playing cards with the LLM and a camera. Have fun with it!"