OzoBlockly Editor Challenge

ORA: The Tetris Kitting Challenge

Student Guide

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

Tetris blocks, technically known as *tetrominoes*, are geometric shapes composed of four square segments connected at their edges. There are seven distinct shapes in total, though this challenge focuses on a specific set of four. Since the game's origin in the 1980s, Tetris has become a global phenomenon because it tests a player's spatial awareness and logic. In this lesson, we are taking that digital logic and bringing it into the physical world, requiring you to think like both a programmer and a geometrician as you link ORA with some Tetrises.

In this challenge, your mission is to have ORA pick up four unique Tetris blocks located at various angles off to the side of the work area and assemble them into a perfect 4x4 square grid. While you are provided with the final orientation of each Tetris block for the 4x4 square, the initial orientation of each piece is intentionally mismatched. To succeed, you will need to master the concept of *yaw*, the rotation of an object around its vertical axis. Think of yaw as the way a car turns left or right on a flat road, how a spinning top rotates, or even how your head rotates when you signal "no" by rotating your head to the left and right. Because each Tetris block begins at a different angle from its final destination, you must calculate the precise change in yaw required to transition each piece from its starting position to its final fit in the grid.

This exercise is a direct simulation of modern "Bin Picking" and "Kitting" operations found in smart factories worldwide. In automotive and electronics manufacturing, robots are frequently tasked with identifying parts that arrive on a conveyor belt in random or skewed orientations. The robot must use its 6-axis flexibility to pick the part and adjust its yaw in mid-air so that it is perfectly aligned for installation into a chassis or a shipping container. By mastering ORA's rotational movements to solve this Tetris puzzle, you are practicing the same high-level spatial reasoning and robotic path planning used by engineers to automate complex assembly lines across the globe.

You are encouraged to view the accompanying video that shows ORA fulfilling the task required by this challenge. The 8½" x 11" black-and-white printed paper shown along with the four colored printed Tetrises in this video are provided for you with this challenge.

General Procedures

Figure 1 contains four images that we now use to explain how to proceed step-by-step in this challenge:

1. Your challenge makes use of four of the seven tetrominoes. These four are commonly referred to as L, J, I, and S, resembling those four letters of the English alphabet. They keep these names regardless of how they might be rotated about axes perpendicular to the page.
2. This 8½" x 11" sheet is taped with double-stick tape to ORA's Quadrant I, positive *x* and positive y. The left shows the initial orientation for each of the four Tetrominoes. The right side shows the Tetrominoes after they have been placed into a 4x4 square. The black dots on the left are where ORA picks up each Tetris with the suction gripper. The black dots on the right are where ORA places the Tetras after adjusting the yaw as necessary.
3. Before starting the OzoBlockly Editor program, colored cutouts of each of the tetrominoes are manually placed on the corresponding figures on the left side of the black-and-white figure.
4. After ORA has completed the task, the tetrominoes have been placed on the 4x4 grid at the right side of ORA's black-and-white workspace.

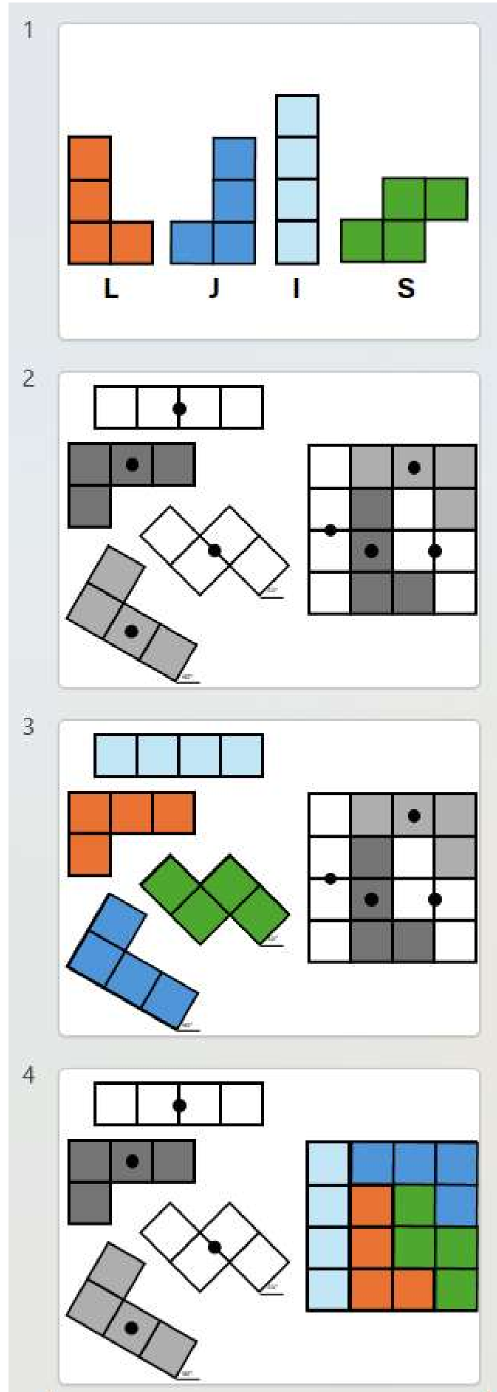

Figure 1

Figure 2 shows the black-and-white printed workspace that has been taped to ORA's Quadrant I. The colored Tetromino cutouts are being manually placed in their locations on the left side of the ORA workspace.

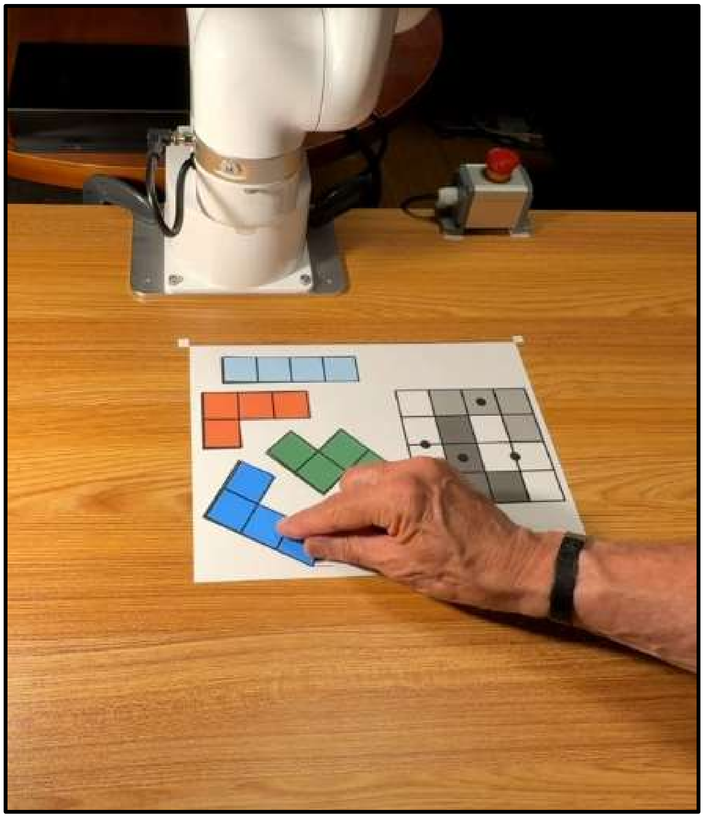

Figure 2

Two <u>VERY IMPORTANT</u> Notes Regarding Safety

1. You should include the Ozobot Editor block shown in Figure 3 at the beginning of your program, perhaps right after the *Program start* block or in a function where you initialize a number of things required by your solution to the challenge. Why? When ORA is moving at a slower speed, say 20% or less, you have more time to prevent an impending collision. Of course, this also allows a person more time to move out of the way of ORA's moving arm!   This is particularly important throughout the development process while creating your program, when you are more apt to make errors in coordinate values in *move to coordinates* blocks.

Figure 3

2. Make sure that the **Emergency Stop Button** is within easy reach. It might even be a good idea to have one student with his or her hand above the button when the program is started, in the

event that it becomes necessary to instantly terminate all motion of ORA's arm and cancel any commands.

Are you ready to bring digital logic into the physical world? This challenge puts you in the driver's seat of a modern smart factory. By mastering yaw and spatial reasoning, you will program ORA to assemble a perfect Tetris grid. Practice the same high-level robotics used by engineers worldwide to automate complex assembly lines!