OzoBlockly Editor Challenge

Precision Stacking – ORA Builds a 2D Pyramid

Student Guide

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

Stacking cubes into a 2D pyramid versus a simple vertical column presents distinct engineering challenges for a 6-axis cobot like the Ozobot ORA. Column stacking is structurally simpler to program, requiring repetitive Z-axis movements and consistent X-Y coordinates. However, this simplicity comes with significant instability. A single misalignment or vibration can cause the entire tower to topple due to its high center of gravity and lack of lateral support.

In contrast, a pyramid structure offers superior stability because its wide base and interlocking layers distribute weight evenly, lowering the center of gravity. While physically more robust, programming a pyramid is computationally more complex. You will need to calculate offset coordinates for each subsequent layer and manage a precise pick-and-place algorithm that accounts for the diminishing number of blocks per row. This forces the robot to execute more varied movements across all axes, making the pyramid a far better test of precision programming and spatial reasoning than a fragile, linear column.

Applications include palletizing shipping containers, where uneven loads must survive transit without tipping, or construction automation, where bricks or blocks must be interlocked for load-bearing strength. Mastering pyramid algorithms will teach you not just about placement, but about structural physics and robust system design.

Retail displays frequently stack items like fruits, canned goods, or drinks in pyramids. This creates an attractive, abundant appearance that draws customers. It also provides a stable and efficient way to display multiple products, encouraging sales and optimizing space in stores like supermarkets.

Challenge Details

You will be using the ORA vacuum gripper to complete this challenge. For your 2D pyramid building project, a vacuum gripper offers significant advantages over traditional claw-like grippers. First, it handles blocks gently, preventing scratches or dents since it picks them up with soft suction cups from the top. This method is crucial for pyramids, allowing you to place blocks perfectly flush against each other without needing space for mechanical fingers. When releasing a block, the suction cuts instantly, ensuring precise placement without shifting, which means straighter, more stable pyramid layers. This simplicity in operation makes it easier to program your robot, helping you build a professional-looking pyramid efficiently and accurately.

What material is best for the cubes that you will be stacking? For picking up cubes with a vacuum gripper, acrylic is probably the best material. Acrylic is inherently non-porous and typically has a very smooth, consistent surface. These characteristics are crucial because they allow the vacuum cup to achieve and maintain an exceptionally strong and reliable airtight seal, preventing air leakage and ensuring maximum suction force. Painted wood is a good second choice as the paint reduces porosity and smooths the surface, but it still introduces more variability than acrylic. Unpainted wood is the least suitable due to its high porosity and rough texture, which severely hinder the creation of an effective vacuum seal by allowing air to leak through the material and around the cup's edge.

Acrylic cubes are likely to be more expensive than painted or unpainted wood cubes. All three types are available through Amazon. One-inch Acrylic cubes, commonly available, are used throughout our discussion of this challenge.

Figure 1 shows a 2D pyramid with five layers of cubes that was constructed by ORA using an Ozobot Editor program similar to the one that you will be designing in this challenge. Notice the following:

- Tight packing of the cubes
- Precision placement without shifting of cubes
- Staggering of layers of cubes by half the width of a cube, so that each cube rests on two cubes directly below it.
- Low center of gravity

Figure 1

The Challenge Setup

Figure 2 shows the setup for this challenge. Cubes are placed into the 3D-printed cube holder on the left one-by-one by a person as needed by ORA. The cube holder has been designed to hold 1" cubes with very little wiggle room and is attached to the table by strong double-stick tape. It is important that each cube be in precisely the same location when ORA moves over to pick it up and that there is no shifting of the cube's position when grasped by the vacuum gripper. The cube holder provides for this capability.

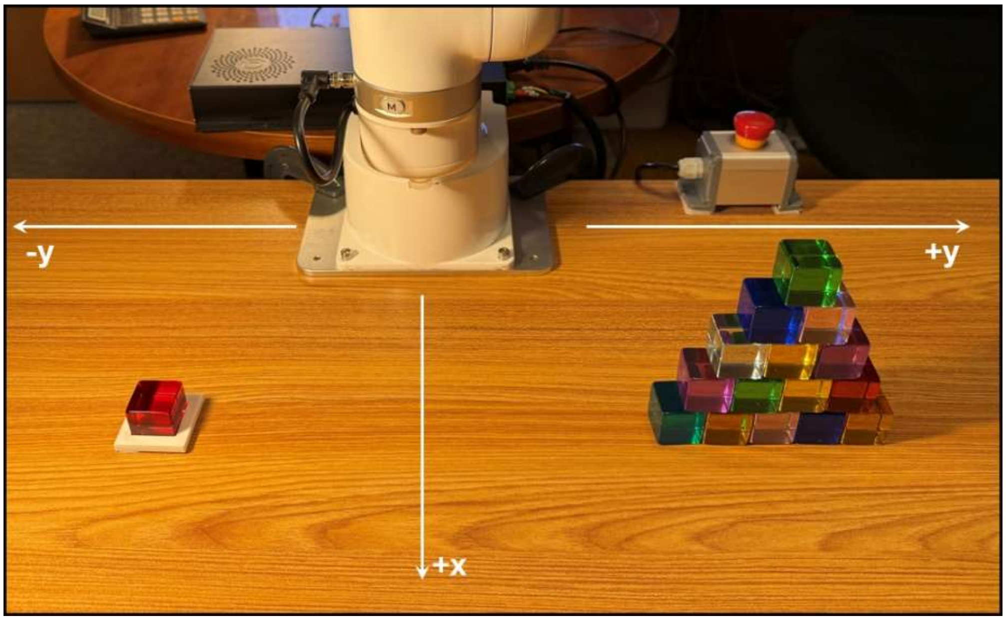

Figure 2

Cubes are placed at the appropriate pyramid location by ORA from left to right in each row. Notice that all cubes in the pyramid have the same *+x* coordinate. The *y* coordinates increase by the width of each cube with each successive cube in a row. ***In might be a good idea to increase the y coordinate by the width of a cube <u>plus one millimeter</u>***. Doing this will lessen the chance of the edge of one cube hitting the edge of the next cube when the cube is being lowered into position by ORA. Turning the vacuum OFF works well when the bottom of the cube held by ORA is 1 mm from the table top.

Each successive row of cubes has one less cube than the row below and each row is staggered to the left by one-half of the width of a cube, making each cube rest on two cubes below it. Each row also has a *+z* coordinate that is the width of a cube greater than the row below.
The number of rows in the pyramid should be initialized in a variable, and your program should build a pyramid with that number of rows. The maximum number of rows allowed should be five. This means that the maximum number of cubes you will need is 1+2+3+4+5 = 15. See Figure 3.

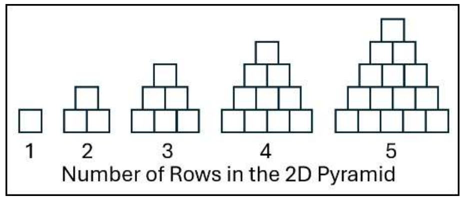

Figure 3

In order to gain a better feeling for the result expected from this challenge, please view the 20-second time-lapse video that accompanies this challenge lesson.

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

1. You should include the Ozobot Editor block shown in Figure 4 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 with the table or cubes that have already been placed by ORA. 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 4

2. Make sure that the **Emergency Stop Button**, shown in Figure 5, 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.

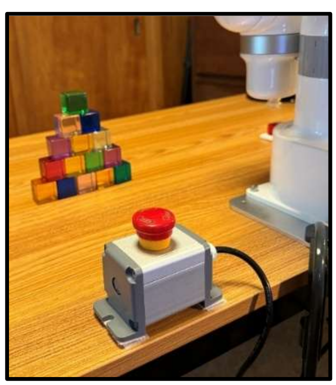

Figure 5

Step into the future of engineering with this OzoBlockly Editor Challenge. Forget simple vertical stacks. You'll command the Ozobot ORA to construct a robust 2D pyramid, a task demanding unparalleled precision and computational prowess. This isn't just block-stacking; it's an immersive dive into advanced robotics, requiring you to master intricate coordinate calculations and meticulous pick-and-place algorithms.

You'll grapple with core engineering concepts like center of gravity and structural integrity, understanding why a pyramid stands strong where a column would falter. Imagine the thrill of seeing your logical programming translate into physical reality, guiding ORA's vacuum gripper to flawlessly interlock each cube. This challenge will not only hone your coding and spatial reasoning but also equip you with invaluable skills for a world increasingly reliant on automated, intelligent systems. Are you ready to build something truly stable and showcase your programming genius?