OzoBlockly Editor Challenge:

Precision Stacking – ORA Builds a 2D Pyramid

Grades 9-12

Teacher Guide

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

This OzoBlockly Editor Challenge tasks students with programming a Ozobot ORA cobot to construct a stable 2D pyramid. This intricate task demands superior programming precision and spatial reasoning, contrasting with simpler, less stable column stacking, and offering a significantly more complex and rewarding engineering project.

The challenge's core involves mastering offset coordinate calculations for each layer and managing precise pick-and-place algorithms for diminishing block rows. Students will execute varied movements across all axes, providing a robust test of their coding and problem-solving skills. Crucially, emphasis is placed on safety. Students are advised to operate ORA at reduced speeds during programming and keep the Emergency Stop Button readily accessible to ensure a secure learning environment.

Students will utilize ORA's vacuum gripper for gentle, precise cube handling, essential for flush placement. The challenge also introduces material science, highlighting non-porous acrylic cubes as optimal for strong vacuum seals. Real-world applications, from palletizing to construction, underscore the relevance of mastering these algorithms. This challenge extends beyond basic programming, offering valuable insights into structural physics and robust system design, fostering a deeper understanding of robotics and engineering.

If you have not already done so, you should now read through the *Student Guide* for this challenge for details on what is expected from the students and view the 20-second accompanying time-lapse video.

Share Code for the Solution – ani3zmk
Share Code for the Solution WITH Comments—mu86p4m
Link to 3D STL File - Cube Holder

Example Solution to this Challenge

The solution to this challenge provides the opportunity for your students to develop an OzoBlockly Editor program that is modular, dividing the program logic into functions each having a specific purpose. Images of the modules follow with detailed comments explaining the purpose of the

Ozobot Editor blocks that make up each module. A separate included PDF file also contains images of these modules, in the event that you want to share them with your students.

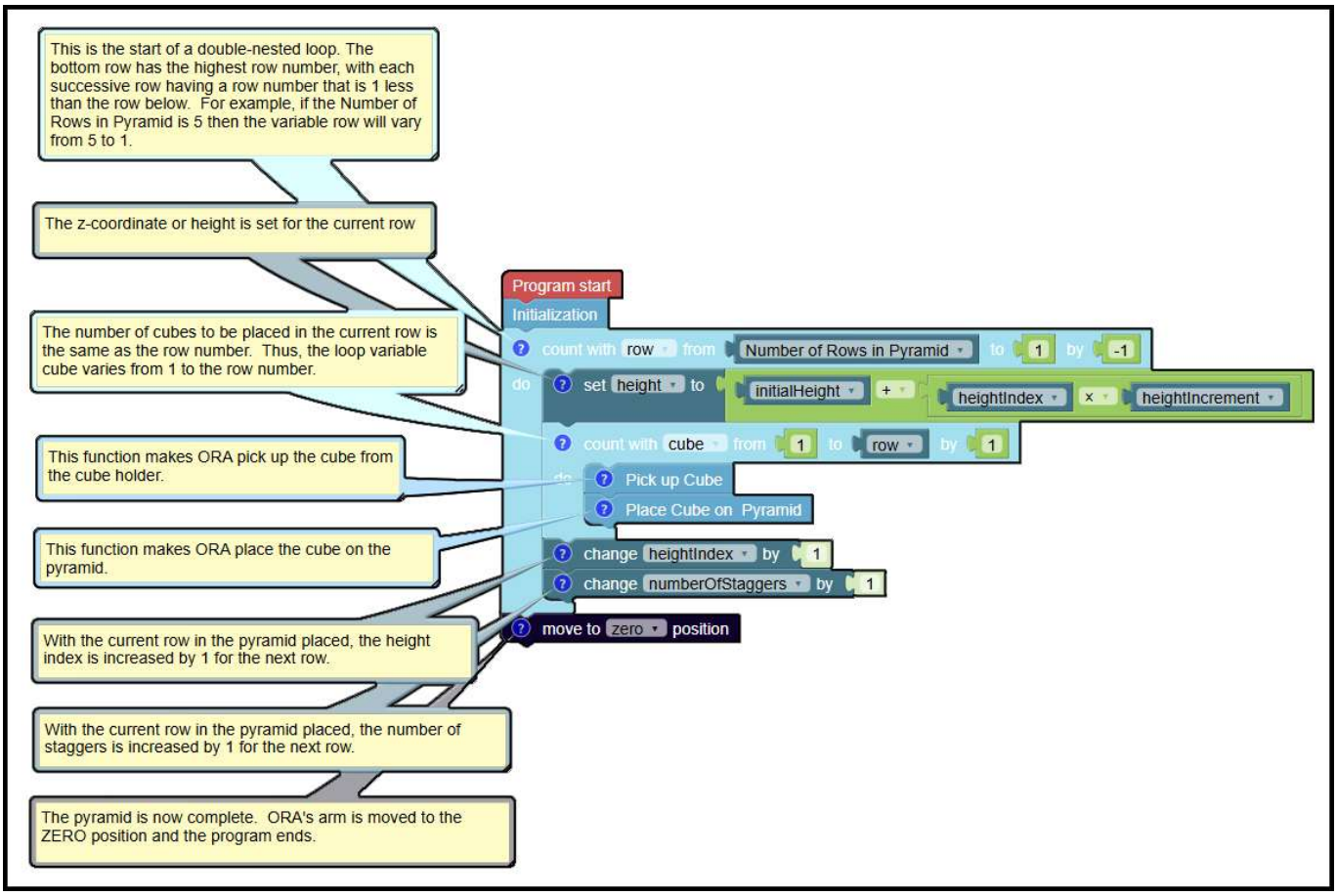

Figure 1

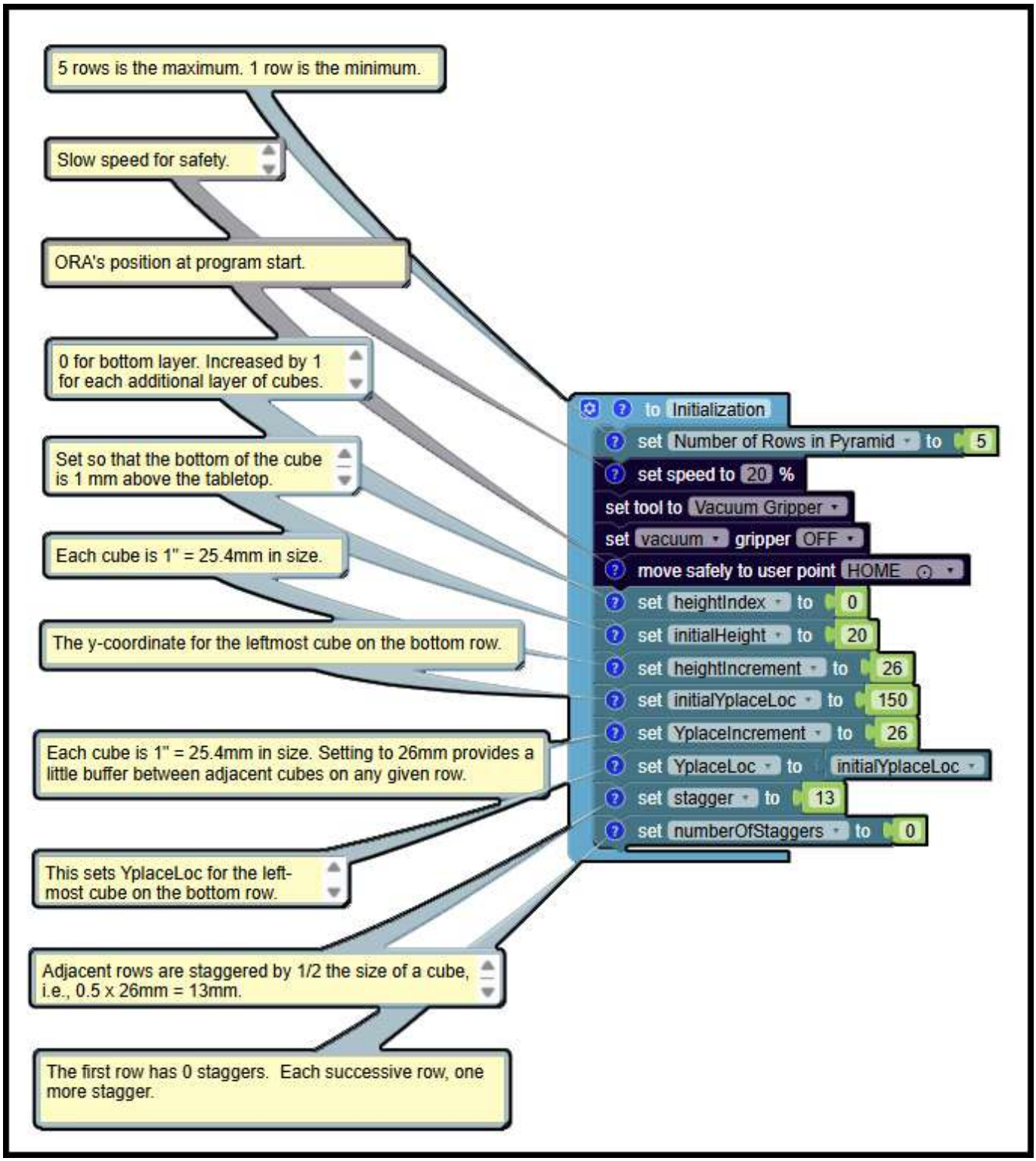
5 rows is the maximum. 1 row is the minimum.
Slow speed for safety.
ORA's position at program start.
0 for bottom layer. Increased by 1 for each additional layer of cubes.
Set so that the bottom of the cube is 1 mm above the tabletop.
Each cube is 1" = 25.4mm in size.
The y-coordinate for the leftmost cube on the bottom row.
Each cube is 1" = 25.4mm in size. Setting to 26mm provides a little buffer between adjacent cubes on any given row.
This sets YplaceLoc for the left-most cube on the bottom row.
Adjacent rows are staggered by 1/2 the size of a cube, i.e., 0.5 x 26mm = 13mm.
The first row has 0 staggers. Each successive row, one more stagger.
to Initialization
set Number of Rows in Pyramid to 5
set speed to 20 %
set tool to Vacuum Gripper
set vacuum gripper OFF
move safely to user point HOME
set heightIndex to 0
set initialHeight to 20
set heightIncrement to 26
set initialYplaceLoc to 150
set YplaceIncrement to 26
set YplaceLoc to initialYplaceLoc
set stagger to 13
set numberOfStaggers to 0

Figure 2

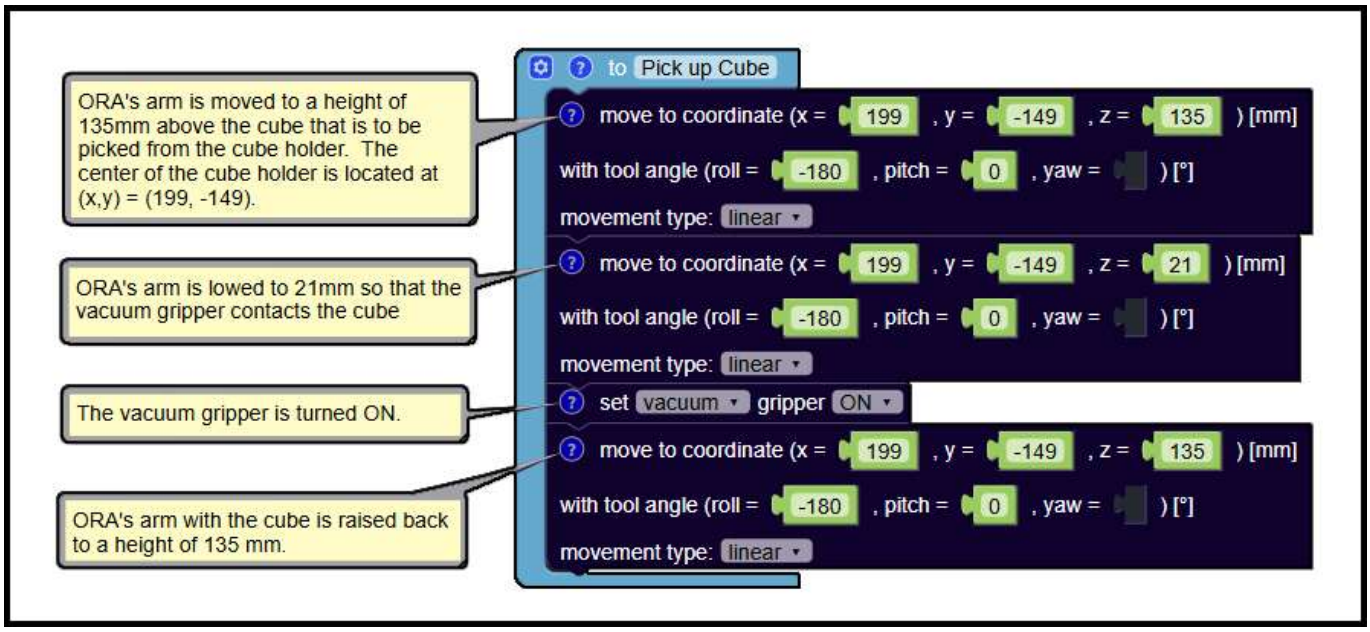

Figure 3

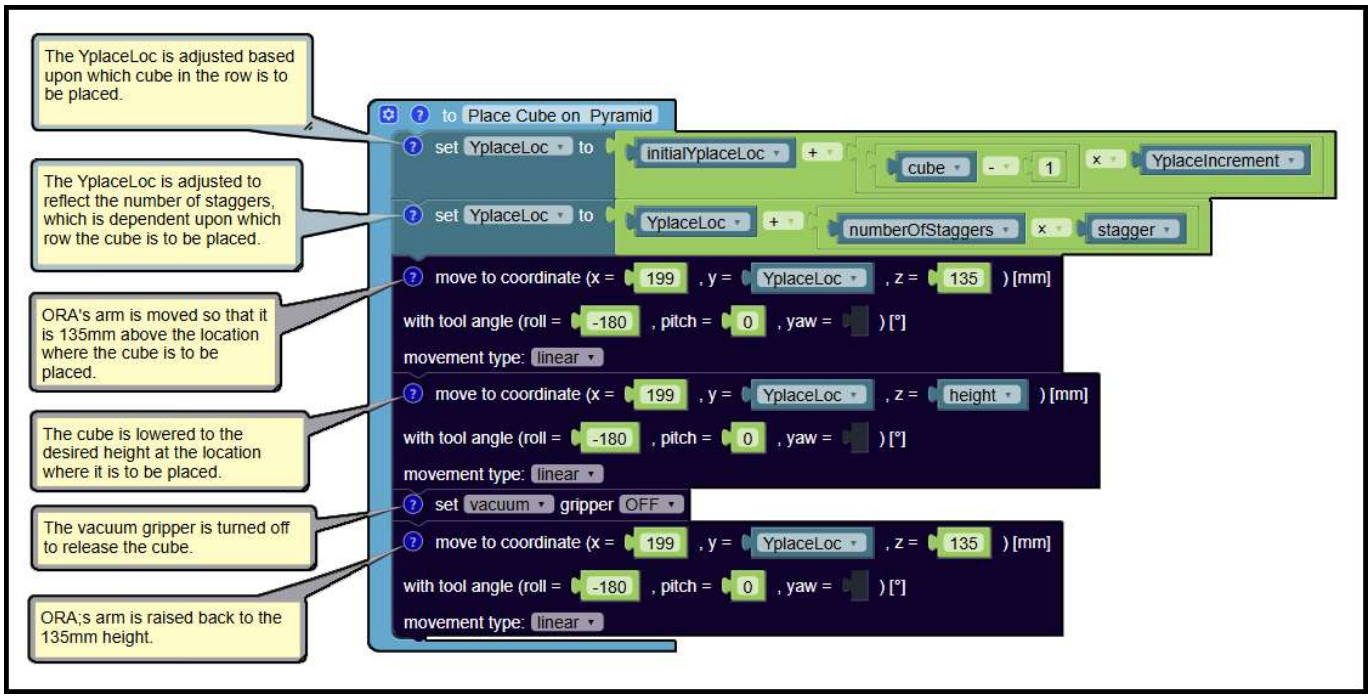

Figure 4

ISTE Standards (International Society for Technology and Education)

Empowered Learner 1.1.d: Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

Knowledge Constructor 1.3.d: Students build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.

Computational Thinker 1.5.c: Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.

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.