Efficiency and Types of Movement:

Name:_____

Lesson 10

Date:_____

Linear - a type of movement where the end tool moves to a point in a straight line.
Joint - a type of movement where the end tool moves to a point through the fewest movements.
Point - a user-defined and recallable position for the ORA.
Efficiency - measured in cycle time, the time taken to complete a product from start to finish.

Engage

You will observe two different programs on the ORA. Compare and contrast the two programs.

Compare	Contrast
Answers may vary but will likely include that ORA visited A1, D1, D4, and A4 in that order.	Answers may vary but should include that the ORA, when running the linear code, moved from quadrant to quadrant in a controlled fashion and with the end tool moving in a direct path. When running the joint code, the end tool path included more arcs and twists.

Explore

You will program ORA to move clockwise from A1 to D4 while avoiding red cylinders placed randomly in your path. Plan out your code and run it using linear motion. Then, run the same code using joint motion.

Which method knocked over the fewest wooden cylinders?
When programmed with the same code and switched from linear to joint, the linear is likely to have knocked over fewer cylinders.

Which method was faster?
The joint method should complete the task faster.

What does this exercise tell us about joint and linear movements?
Answers may vary but are likely to include that each movement type has its own set of special use cases.

Efficiency and Types of Movement:

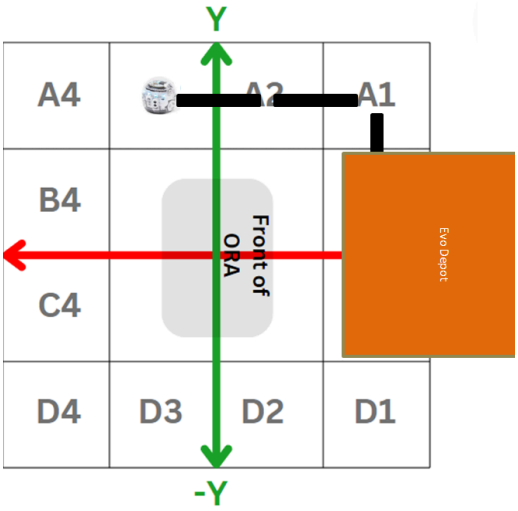
Name:_____

Lesson 10

Date:_____

Elaborate

In this challenge, students will use ORA and Evo concurrently while utilizing custom-created points.



Task Requirements:

1. ORA must start in the Initial Position.
2. Students must run ORA and Evo concurrently.
3. Evo will follow the line from A3 into the "Evo Depot" and begin to flash the top lights.
4. Students will manually move ORA to create custom waypoints to guide the robot arm into the box and down to pick up the Evo.
5. Students will then return Evo to A3 to continue following the line.
6. ORA must not touch the "Evo Depot" Box.
7. ORA must finish in the Initial Position.

Pseudocode	Code Blocks
Answers will vary but should include: Setting the tool to finger grippers Moving to initial position (beginning and end) Moving to user points Opening/closing the finger grippers Concurrency between Evo following the line and ORA moving back to pick it up from the 'Evo Depot'	Answers will vary but should include: Set tool to Finger Gripper block Move to initial position block Several "move safely to user point" blocks Concurrency blocks

After testing, was your code successful? If not, what changes will you make to accomplish the task?

Answers will vary

Efficiency and Types of Movement: Lesson 10

Name:_____

Date:_____

Evaluate

Please answer the following prompts.
Remember to include your **name** and **date** on this paper!

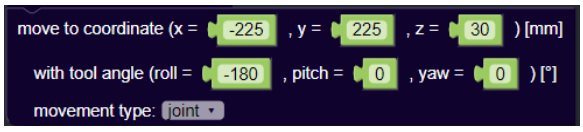
1. Based on the data you collected during the Explore section, which type of movement (linear or joint) was more predictable?

The linear movement type should be more predictable.

2. Based on your answer above, which movement type should you choose to create a code with the most efficient ORA movement? Were there other considerations to take into account when choosing a movement type?

Answers will vary, but since each movement type has its own set of special use cases, it can be a case-by-case assessment on whether to use linear or joint movement.

3. What is the purpose of creating custom points, and how is using them different than simply coding for those movements with a **Move to Coordinate** block?



vs.

Answers will vary, but the **move to coordinate** block is larger, which results in a longer code. It is much more compact to use several **move safely to user point** blocks. The custom points make it easier to recall user-created positions. On the other hand, the **move to coordinate** block has an advantage in that you can easily see the X, Y, Z, R, P, and W values.