Think Like a Robot Activity (30-40 minutes)-Zero Robotics

1. Ideally, after students complete the “You Have Nothing to SPHERE” activity, ask them to put their donut hole models to the side to be used later. This activity is intended for students to get a better idea of what it takes to program a robot and to understand how to write code that properly executes according to the robot’s orientation and position.
2. Construct a maze or “Obstacle course” for students to traverse by foot, using tables, pieces of paper on the floor, books, etc. for the perimeter and “track.” Make sure the “arena” is safe for students to navigate by having no rough edges, no slippery surfaces, and adequate horizontal and vertical space for a student to clear.
3. Ensure students understand the concepts of “attitude” vs. “direction” on the ISS and in the IDE. Students cannot use “sensors” for this initial activity, though later activities can be done with this concept in mind. For example here, if the student (robot) were to accidentally hit a wall, they should immediately stop, as they don’t have the sensors (with their eyes closed) to tell if they are too close to a wall. Remind students that a robot (i.e. them) cannot function without its programming.
4. Students should be split up into groups of two, with one student acting as the robot (who will traverse the maze), and the other as the programmer. Make sure the student pairs are able to cooperate with each other and are capable of working together.
5. Ask the “robot” to grab their donut hole model and position themselves at the start of the maze. If desired, the robot can close his/her eyes to further emphasize that they cannot move without the help of their programmer. Have the “robot” orient their donut hole model correctly to match the orientation of the walls in the “Simon Says” activity. They will need to maintain their model’s orientation throughout the entire maze in order to successfully “complete” it.
6. The goal is for each group to traverse the maze using as few steps (akin to instructions or lines of programming) as possible. Alternately, after each robot makes it to the end, change the maze up a little bit to make it more difficult for the next team to traverse, emphasizing the greater complexity of the programming needed in order to successfully perform more complex tasks. If time, have each group’s “robot” and “programmer” switch roles to see the differences and intricacies each role brings to the challenge.
7. After all of the groups have traversed the maze, bring them together for 5-10 minutes of discussion and reflection.
8. Ask students questions about what happened. *Sample questions:* What parts were easy and difficult? What was the hardest part? Do you think this is a realistic way to think about a robot's programming (specifically, the SPHERES programming)? Why or why not? ADVANCED: How could the programming and/or actions be improved with the use of sensors or other additional tools such as loops, conditional statements, and parallel (simultaneous) programming? Regardless of their experience and/or age level, it’s almost always a challenge for the students to do this successfully the first time and, thereafter, program their robot correctly the first time. Plus, a key concept is that programming has to be highly detailed, structured, well-organized, and very precise in order to be executed well.