 Activity

|  |  |
| --- | --- |
|  | Cube Pusher Code the BaseBot to push cubes out of the square! |

# Step by Step

1. [Build the BaseBot](http://link.vex.com/iq/builds/basebot/iq-2nd-gen-basebot) and attach an Optical Sensor pointing down from the front plate. Open the BaseBot (Drivetrain 2-motor) template in VEXcode IQ and configure the Optical Sensor.
2. Use red tape to create a 3’x3’ square using the black lines on the Tiles as a guide, and place 3 cubes inside the square, as shown in the image above. The color of the cubes does not matter.
3. How can we use the Optical Sensor to keep the BaseBot in the square, as well as pushing the cubes? The Optical Sensor will detect the red tape as the edge of the square. When the BaseBot detects red with the Optical Sensor, it will back up and turn right, in order to push more cubes.
4. Place the BaseBot at the starting location shown in the image above, and build the code in the image to the right. Input parameters in the [Drive for] and [Turn for] blocks to determine how far the BaseBot should back up and turn. How far should the BaseBot move backwards? How will changing the degree of turn affect the path the BaseBot takes?
5. Download and run the project to test it! Change the parameters in the [Drive for] and [Turn for] blocks to keep the BaseBot in the square as needed.

|  |  |
| --- | --- |
| ‘LEVEL UP’  * **More Cubes -** Add additional cubes inside the square for the BaseBot to push out. * **Cube Crasher Race** **-** How fast can you push all the cubes out? Time yourself or practice with a friend. | Pro Tips  * The [Set drive velocity] block can be used to increase or decrease the BaseBot’s velocity. The default velocity is 50%. |

**Standard:** **CSTA (** 2-AP-12) Algorithms and Programming- Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.