Dex the robot’s obstacle course

Planned Course (Orange is Dex’s rout)

**RED:** STEP 1

This is the first post that Dex needs to navigate around. It’s purpose is for Dex to go around, then continue along the planned route. The challenge this turn created for Dex was to completely navigate around the post using multiple turns and align itself for the next part of the course.

**YELLOW:** STEP 2

This is the second point Dex is required to navigate. This is meant for Dex to go nearly completely around, utilizing a series of turns and forward movement. Dex also needs to go in-between step 1 and Step 3 (closest post) to be in the correct position for the next step.

**BLUE:** STEP 3

This third step in the course consists of two posts. Dex has to navigate through these two posts as well as turn about the far post. Dex is required to do various turns and movements in order to achieve this without running into anything. This will lead Dex to the next section.

**Purple:** Step 4

This step houses a bump that Dex needs to go over. There is no extreme incline due the limited actions that a base bot is able to perform, but it still shows how even the most basic of robots can still traverse miner terrain difficulties.

**Pink:** Step 5

There are two parts to step 5. There is first a ball that Dex needs to grab, he then speeds up so that the ball will have greater momentum. Dex will book it to the second part of this section, the “net”, where he will stop just before it. This will let the ball roll forward into the “net.” Once he and the ball have departed, he backs up and turns, ready to conquer the next challenge.

**Green:** Step 6

This step was initially supposed to be two bumps for Dex to go over. These bumps would have had a higher incline than the first bump in step 4. We had to change this because Dex was not able to go over these and would instead push them. This new tower that Dex needs to push to the finish line shows the strength of Dex during the final stretch of the course.

(We used block code)

myVariable = 0

def when\_started1():

global myVariable

drivetrain.set\_drive\_velocity(30, PERCENT)

drivetrain.drive\_for(FORWARD, 49, INCHES)

drivetrain.turn\_for(RIGHT, 45, DEGREES)

drivetrain.drive\_for(FORWARD, 21, INCHES)

drivetrain.turn\_for(RIGHT, 90, DEGREES)

drivetrain.drive\_for(FORWARD, 17, INCHES)

drivetrain.turn\_for(RIGHT, 20, DEGREES)

# The previous code was to maneuver Dex around our first obstacle (step 1)

drivetrain.drive\_for(FORWARD, 27, INCHES)

drivetrain.turn\_for(LEFT, 50, DEGREES)

drivetrain.drive\_for(FORWARD, 18, INCHES)

drivetrain.turn\_for(LEFT, 85, DEGREES)

drivetrain.drive\_for(FORWARD, 18, INCHES)

drivetrain.turn\_for(LEFT, 60, DEGREES)

drivetrain.drive\_for(FORWARD, 18, INCHES)

drivetrain.turn\_for(LEFT, 30, DEGREES)

drivetrain.drive\_for(FORWARD, 28, INCHES)

# The previous code was used to move Dex around the stake

# There was multiple turns because the stake was bigger obstacle and harder to move around in one turn

drivetrain.turn\_for(RIGHT, 90, DEGREES)

drivetrain.drive\_for(FORWARD, 36, INCHES)

drivetrain.turn\_for(LEFT, 70, DEGREES)

drivetrain.drive\_for(FORWARD, 25, INCHES)

drivetrain.turn\_for(LEFT, 45, DEGREES)

# This allowed Dex to move around the far post and align with the next obstacle

drivetrain.set\_drive\_velocity(75, PERCENT)

drivetrain.drive\_for(FORWARD, 55, INCHES)

drivetrain.set\_drive\_velocity(30, PERCENT)

drivetrain.drive\_for(REVERSE, 13, INCHES)

drivetrain.turn\_for(LEFT, 90, DEGREES)

# This is where Dex becomes the GREATEST soccer player in the world (and does a sick wheelie)

drivetrain.drive\_for(FORWARD, 102, INCHES)

# The previous line was the final stretch of the obstacle course where Dex will move the stack of rings

when\_started1()

Dex has completed his first turn in the course

This is when Dex goes over the bump and grabs the ball

Dex will push the stack of rings

**Reflection:**

This was good instruction on how VEX EXP works as well as how to interact and operate the components. We had initially wanted to make smoother turns and were able to figure out how to make that possible. However, due to the limited amount of data our sad little middle school brains can handle, we decided to keep things simple and stayed with the original method of turning. Some challenges we faced were aligning the turns and getting the correct distances so that Dex would be able to accomplish the course. We were also required to make sure Dex started in the correct spot, otherwise movements wouldn’t align with the obstacles, and we’d redo our code. This would lead to us constantly having to change our code and obstacles because of where we started. These changes may have hurt our brains, but we ended up learning from them in the end.