Challenge

Students will design, program, and build a robot that drives around in town while avoiding collisions and staying on the roads. The robot should turn around when it reaches the outside perimeter. All the class robots will run at one time.

Materials Needed

Each pair of students will create one robot.

- Use one of these sets:
  - TETRIX® PRIME Programmable Robotics Set (44321)
  - TETRIX PRIME Dual-Control Robotics Set (44322)
- Items to create challenge field: painter's tape
- Engineering logbook

Objectives

By the end of the lesson, students will be able to:

- Design and build a challenge field.
- Build a robot within the constraints to meet the challenge.
- Write the steps and create a program for the robot that meets the challenge.
- Test and refine the robot program and design.
- Demonstrate the effectiveness of the robot to meet the challenge.
- Reflect on and share the challenge and its real-world applications.

Activity

Robots in Town Challenge

Difficulty

Intermediate

Class Time

Six or more 45-minute class periods

Grade Level

- Middle school
- High school

Learning Focus

- Engineering problem-solving
- Robot assembly
- Computer science
- Statistics and data collection

Step 1: Introduce (15 minutes)

- Share, define, and refine the challenge. Document this information in the engineering logbook.
- Write the challenge in your own words. Record the constraints you should follow, the materials that can be used for the solution, and what the testing field will look like. Discuss the constraints and materials that are allowed.

Step 2: Brainstorm (30 minutes)

- Brainstorm ideas to solve the challenge. Create quick sketches and describe solutions to the challenge.
- Considerations for your design:
    - The entire community will need to agree to some driving laws for the town. For example:
        - Accidents come in three types:
            - Collisions – How severe is it? Who is at fault? What is the penalty?
            - A wheel that goes outside of a street line has had an accident.
            - Driving off-road – how far is too far? What is the penalty?
        - Speed limits
            - Are they different on different roads?
            - What are the penalties for speeding? For going too slow?
    - The entire community will have to decide on what data to collect, how to collect it, and how to analyze it. For example:
        - The number and types of accidents
        - Community driving habits
        - The size of the car in terms of navigation and maneuverability

Step 3: Set Up (15 minutes)

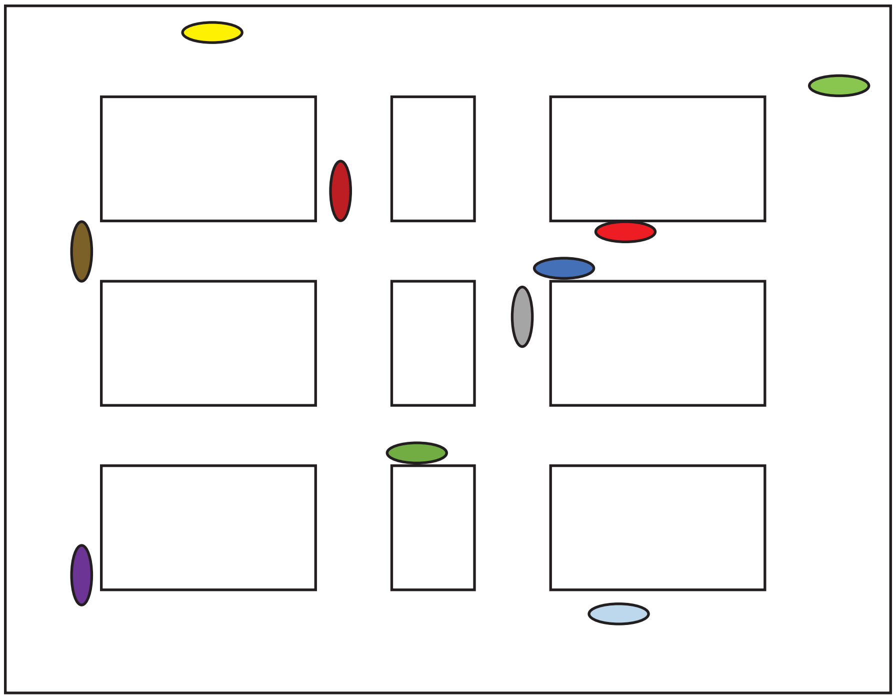
- Build the challenge field following the pictured guide.
    - Create several intersecting streets in a small town. Use black lines for the streets.
        - Intersecting streets should be about 80 cm wide.
    - Place a double-black line to represent the perimeter boundary of about 5 m by 5 m or greater.
    - Use about 5 m by 5 m of floor space, or you could use the entire classroom with desks located as if they are buildings.

Constraints

The team's robots must:

- Contain parts from only one set.
- Measure less than 30 cm x 40 cm x 40 cm.
- Use an Ultrasonic Sensor to avoid other vehicles.
- Use a Line Finder Sensor to stay on the streets.
- Turn around when the robot reaches the perimeter.

Possible Challenge Field



Step 4: Plan (30 minutes)

- Before building, think about the potential design of the robot and draw or record ideas in the engineering logbook. Consider the following:
  - Drivetrain for speed and control
  - Robot chassis for size
  - Location and orientation of the Line Finder Sensor
    - For cars that drive on the right side of the street, the sensor should be located on the outside right side of the robot.
  - Location and orientation of the Ultrasonic Sensor
    - Should it be on the front of the robot or the top?
    - Should it be stationary or rotating so that it looks left, right, and forward?
  - A manipulator to indicate when the robot is involved in a collision
- Create a detailed sketch of your selected solution to the challenge. Label the materials you will use. Write a detailed description of how your solution meets the challenge, constraints, and criteria.

Step 5: Create (45 minutes)

- Design and build the robot. Remember to update the solution in the engineering logbook as the design is improved.
  - **Note:** The creation of the robot could take longer depending on the complexity of the robot solution.

Step 6: Write the Steps (15 minutes)

- Think through the steps or series of actions that the robot will have to complete to meet the challenge. Planning this series of steps is sometimes referred to as creating pseudocode for your robot.
  - Record these steps in the engineering logbook and use them as a guide when operating the robot. Notice that the steps are like writing code for the robot to follow. Make sure the robot performs all the steps required in the challenge.

Sample Steps

1. Drive along a street by following a line.
2. Stop if I encounter another car.
3. Allow the other car to be moved out of the way manually.
4. Continue driving.
5. Stop and turn around if I encounter the perimeter of the town.
6. Signal if I collide with another car.

Step 7: Create the Program (45 minutes)

- When you have completed this process, you are ready to begin programming using your steps as a guide. Remember to track changes in the engineering logbook.
  - When you are coding your robot, it is recommended that you write the code using functions so that each task can be tested and adjusted before it is incorporated into the final program.
- Prepare functions to control your robot, depending upon your solution plan.
- Check each of your functions as you write it to make sure it works as you intend.
- Now, write a test sketch to try them all out.

Step 8: Test (45 minutes)

- Test the solution. Place the robot into the challenge field and press the Start button to execute the code.
- Refine the solution. Adjust the design and code as needed. Document any changes in the engineering logbook.

Step 9: Demonstrate (15 minutes)

- When the robot has been tested and successfully navigates the challenge field, demonstrate its performance in a final test.

Step 10: Reflect and Share (15 minutes)

- Look back at the prototype. How does it compare to the final design?
- Look back at the original steps. How do they compare to the final steps?
- Discuss the original prototype, the final robot code, the solution as implemented, and how this challenge applies to the real world of robot design and programming.

Step 11: Extensions

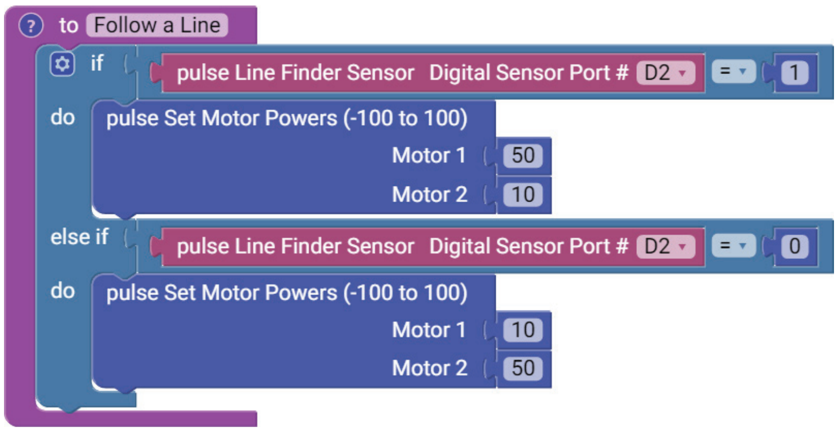
- Commuter Car
    - Add homes for each of the robots and a common work-site parking lot for all the robots. Have all robots start from their homes at the same time and travel to the company parking lot and park. Are there traffic jams anywhere? Does it matter where the home is in relation to work? How long does it take for everyone to get to work? By leaving at different times, could this overall time to get to work be shortened? Does each robot have an assigned parking spot, or is it random? When all robots are at work, can they all make it back home?
- Traffic Jam at Lattes-R-Us
    - Transform the Commuter Car to go to the center of town to pick up a latte at the only morning place in town, then from there, to find its way to work.
- Taxi Bot
    - Add signs so that when the randomly driving robot sees one, its stops to pick up a passenger and takes the passenger to the mall, stops to let the passenger out, then returns to randomly driving around town. Transform this into a game by seeing which robot can take the most passengers to the mall in a set time.
- Mail Bot
    - Transform the Taxi Bot into a mail carrier by adding the task of actually picking up the mail (a small box) at a sign and delivering it to the post office.
- Delivery Bot
    - Transform the Taxi Bot into a delivery truck by adding the task of dropping off a package (a small box) at every sign it finds.
- The City
    - Populate the town with two or more robots from the Commuter Car activity, two from Traffic Jam, two Taxi Bots, one Mail Bot, and one Delivery Bot. Start them all and think like a city planner on how to minimize accidents and traffic jams.

To get you started, here are some sample blocks for the PULSE™ controller with the *TETRIX Ardublockly* software.

Try to incorporate while loops. You will need to program the robot to deal with:

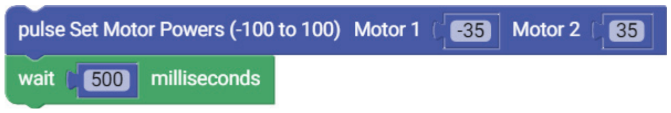
- The double-black perimeter line (Line Finder Sensor)
- The black road lines (Line Finder Sensor)
- Possible collisions (Ultrasonic Sensor)

You could also incorporate the use of functions to name a series of programming steps such as turn left, turn right, go backward, go forward, stay on road, look for perimeter, look for car, or brake.



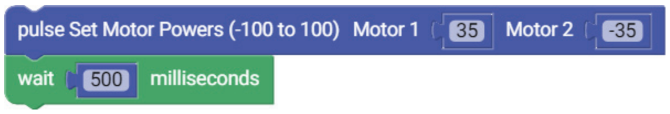
```
to Follow a Line
  if   pulse Line Finder Sensor  Digital Sensor Port # D2  = 1
  do   pulse Set Motor Powers (-100 to 100)
                                     Motor 1  50
                                     Motor 2  10
  else if  pulse Line Finder Sensor  Digital Sensor Port # D2  = 0
  do   pulse Set Motor Powers (-100 to 100)
                                     Motor 1  10
                                     Motor 2  50
```

In this set of blocks, the robot will follow a black line.



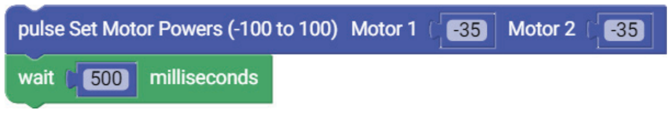
```
pulse Set Motor Powers (-100 to 100)  Motor 1 -35  Motor 2 35
wait 500 milliseconds
```

Turn left



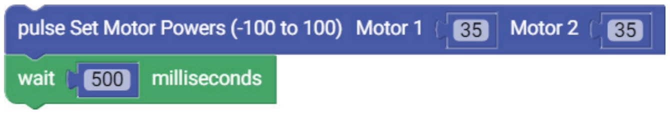
```
pulse Set Motor Powers (-100 to 100)  Motor 1 35  Motor 2 -35
wait 500 milliseconds
```

Turn right



```
pulse Set Motor Powers (-100 to 100)  Motor 1 -35  Motor 2 -35
wait 500 milliseconds
```

Go backward



```
pulse Set Motor Powers (-100 to 100)  Motor 1 35  Motor 2 35
wait 500 milliseconds
```

Go forward