Autonomous Vehicles

December 8, 2022

Lesson Plan Structure

First half is planning and preparation. Second half is testing, debugging, and improving.

Lessons 1/2: Self-parking
Lessons 3/4: Safety
Lessons 5/6: Navigation
Lesson 7/8: Advanced Features

NY State Standards for Computer Science

9-12CT.1 Use computing tools and techniques to create artifacts. [P2] Challenge activities result in the creation of a (simple) algorithmic solution and an accompanying program that implements it.

9-12CT.10 Collaborate in the creation of computational artifacts. [P6] Students work in teams to accomplish tasks.

9-12CT.2 Analyze computational artifacts. [P4] Students perform debugging on their own code, as well as analyze and evaluate others' code and suggested code in Reflection Questions.

9-12CT.1 Use programming as a creative tool. [P2] **Students use programming to solve model challenges based on challenges real robots face.**

9-12CT.2 Develop an abstraction. [P2] **Robots gather information about the world through sensors, which turn physical qualities of the world into digital abstractions. Students must understand and work with this data to develop then implement their solution algorithms.**

9-12CT.1 Use models and simulations to raise and answer questions. [P3] **Students construct and use a "program flow" model of programming itself to understand how the robot uses data to make decisions and control the flow of its own commands.**

9-12CT.6 Develop an algorithm designed to be implemented to run on a computer. [P2] Students develop solution algorithms to each challenge and mini-challenge problem before implementing them as code. Reflection Questions also ask students to evaluate algorithms expressed as pseudocode.

9-12CT.8 Express an algorithm in a language. [P5] **Students develop code to robotics challenges in the EV3 Programming Language.**

Lesson 1/2: Self-Parking (2x45 min)

How do cars park themselves without human intervention?

D/N: (In groups 20 min)

Students will brainstorm ideas on how to program EV3 bases to park themselves. What components and settings are needed to complete the maneuvers?

Students will develop prototypes to self-park their EV3's in three ways:

1. Parallel parking
2. Angular parking
3. Perpendicular parking

Lesson 1/2: Self-parking (continued)

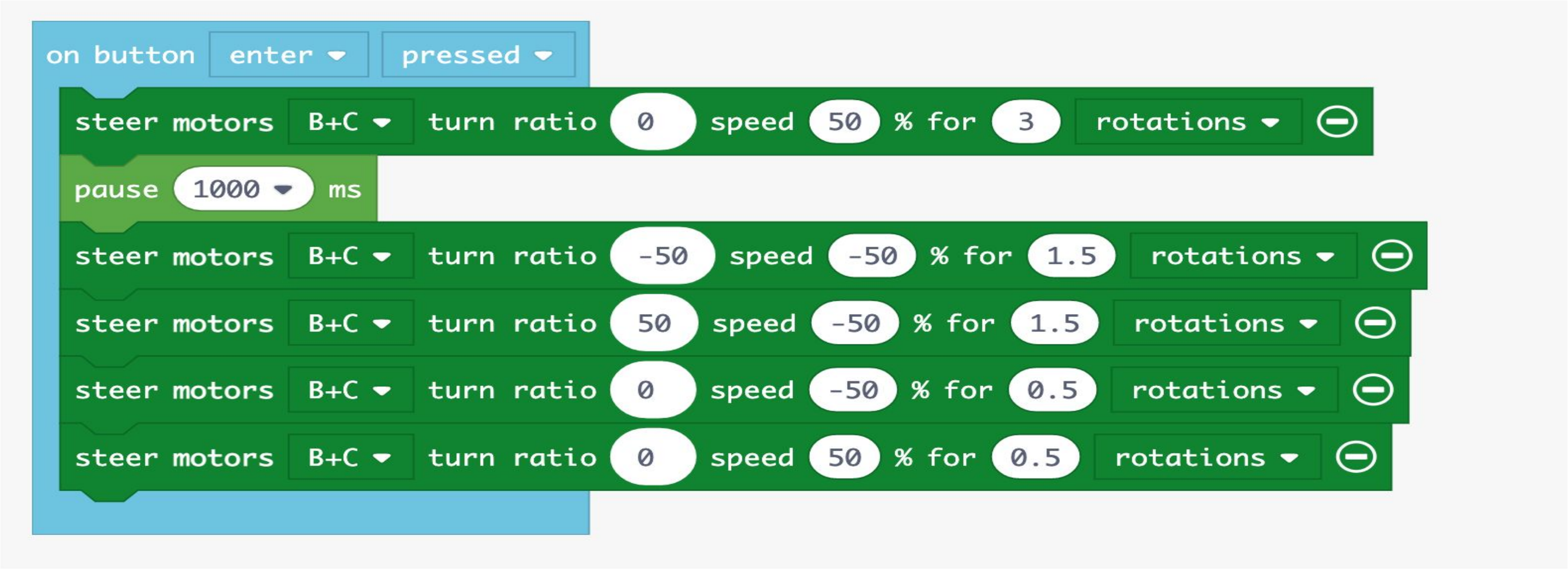
Have students begin testing simple movements first. Record results: (25 min)

1. Forward
2. Backward
3. Simple turns left and right
4. Combination of forward turns and backward turns
5. U turns and K turns

Continue making adjustments to the settings to achieve more precise movements, especially when the vehicle is in tight spaces.

Lesson 2/2: Self-parking (continued)

Sample code for parallel parking:



Lesson 2/2: Self-parking (continued)

Sample video of perpendicular parking robot:

https://drive.google.com/file/d/1OFxDwkIkVN4qKnVnevmfsQfcZq34UwhP/view?usp=share_link

Sample video of parallel parking robot:

https://drive.google.com/file/d/1wCXbk5zjcg1y9SG8ZSVL9IEM1i-xZEYi/view?usp=share_link

Lesson 2/2: Self-parking (continued)

Have students write prototype codes for all three modes of parking. (30 min)

Test each of them, making adjustments to the different settings to improve movements to their vehicles.

Have students take videos of their programs working.

Wrap Up (written response): (15 min)

1. What were some of the challenges your group encountered and what were your solutions?
2. What improvements did you make during the design and prototyping phase? How would you further improve your program?
3. How can your program be used/adapted to real world situations?

Lesson 3/4: Safety (Collision Prevention, 2x45 min)

How can autonomous vehicles avoid accidents and collisions?

Materials: EV3 Base, Ultrasonic sensor

D/N: (Groups 20 min)

Students will brainstorm with their partners on different situations in which accidents can occur and what types of systems cars can use to help avoid collisions with other vehicles, objects, people, etc. Then, students will research the systems that are actually used in today's cars. How do they negotiate traffic? They will submit their work at the end of the period.

Students will then prototype programs for different situations where their robots will detect obstacles and react to them in order to prevent a collision:

1. Detecting object while moving forward or backward
2. Using various speed settings for different stopping distances

Lesson 3/4: Safety (Collision prevention, cont’d)

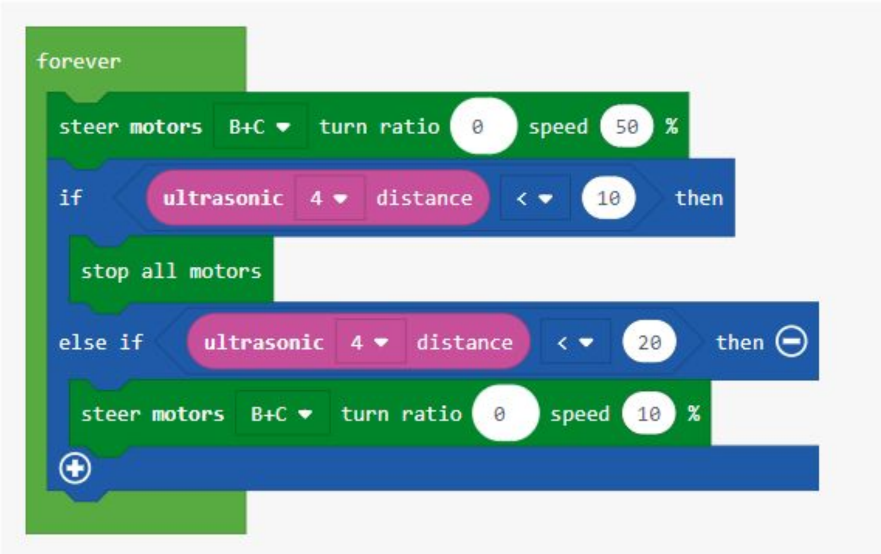
Students will test their prototypes for simple movements using the ultrasonic sensor. They will make adjustments to prototypes to improve their movements (25 min)

1. Should their stopping distances increase/decrease?
2. Are they stopping abruptly or smoothly?
3. Should their robots come to complete stop or slow down and go around the obstacle?

Students will continue working on their prototypes for simple motions and record their results.

Lesson 3/4: Safety (Collision prevention, cont'd)

Sample code blocks for slowing, then stopping when obstacle detected:



Lesson 3/4: Safety (Collision prevention, cont’d)

Sample video of course maneuvers and avoiding collisions.

https://drive.google.com/file/d/1WQwhPHHk_iV5P6SCXpigPg7TE5Ru0mzq/view?usp=share_link

Lesson 3/4: Safety (Collision prevention, cont’d)

Challenge: (35 min)

Make your robot maneuver safely through an obstacle course!

Wrap Up (written response - 10 min)

1. What were some of the challenges your group encountered and what were your solutions?
2. What improvements did you make during the design and prototyping phase? How would you further improve your program?
3. How can your program be used/adapted to real world situations?

Lesson 5/6: Navigation (2x45 min)

How do autonomous vehicles navigate successfully to their destination?

Materials: EV3 Base unit, color (light) sensor, color bricks (simulating traffic lights)

Task: (Groups 45 min)

Have students brainstorm how cars can navigate without human intervention. What systems would they use? How do they work together? How do they negotiate traffic lights? How can we detect if a driver is asleep at the wheel? Submit a written response by the end of class.

Students will develop prototype codes that make the their robots react to different color blocks or road markings when encountered:

1. Stop at red bricks, go at green bricks
2. Stay on path following a color strip
3. Turn in the correct direction when encountering certain colors (decision)

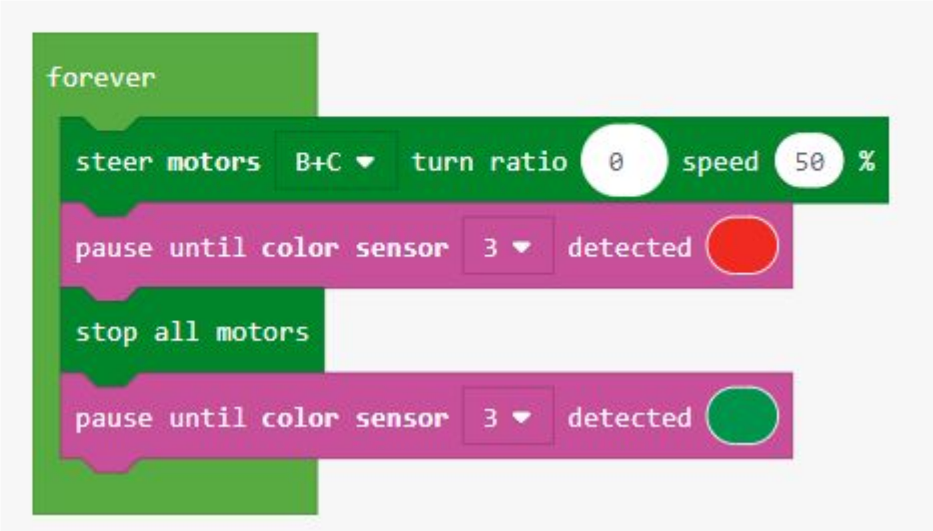
Lesson 6/6: Navigation (continued)

Students will test their prototype codes for the different situations and make adjustments if necessary. (25 min)

1. How well did the robot go/stop at a green/red “light”? Explain
2. How well did the robot navigate the prescribed path? Did it follow smoothly? If not, what adjustments need to be made in the code?
3. Did the robot make the right decisions at turns? What problems did it encounter?
4. What safety features can I implement into the robot if it detects the driver is asleep or falling asleep? A sound alarm or tactile feedback system?

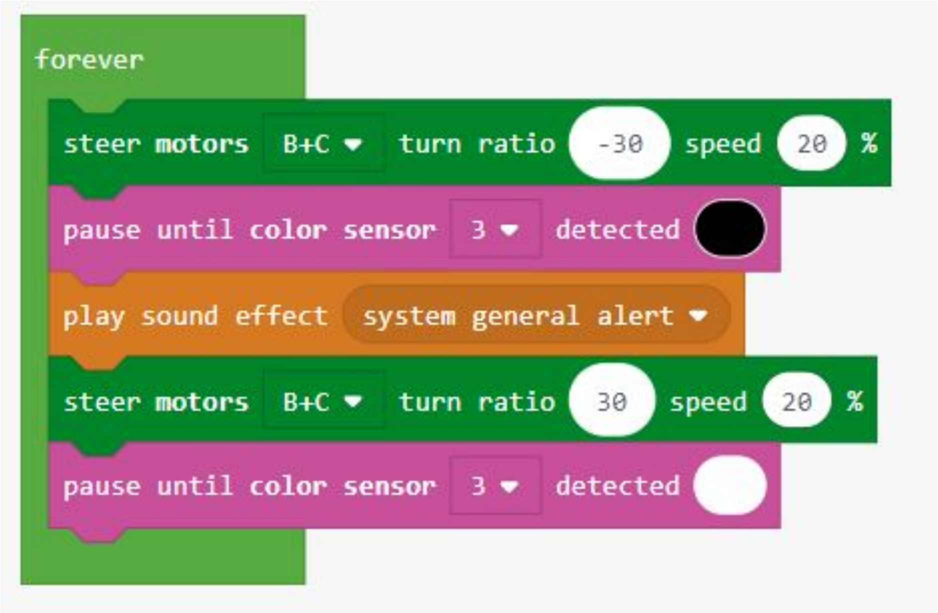
Lesson 5/6: Navigation (continued)

Sample code for traffic light simulation:



Lesson 5/6: Navigation (continued)

Sample code for detecting sleeping driver and correcting path:



Lesson 5/6: Navigation (Continued)

Sample video of robot navigation:

https://drive.google.com/file/d/1XST-n0Tt0pGj7SgfHlOEW3D2-8b0frnm/view?usp=share_link

Lesson 5/6: Navigation (continued)

Wrap up: (20 min)

Have students video their robots and share out with the class.

Wrap Up (written response):

1. What were some of the challenges your group encountered and what were your solutions?
2. What improvements did you make during the design and prototyping phase? How would you further improve your program?
3. How can your program be used/adapted to real world situations?

Lesson 7/8: Advanced Features (2x45min)

How can students investigate advanced features of autonomous vehicles by creating programs so their EV3 robots follow moving objects or navigate more complex obstacle courses?

Planning and Preparation: (45 min In groups)

Have students brainstorm and research more complex features of autonomous vehicles and create algorithms to code these features into their EV3 robots. Allow them to make modifications to their existing codes to complete the more complex tasks. Record all modifications you made and develop prototypes to test during the following class.

Lesson 7/8: Advanced Features (continued)

Test Prototypes, Make Adjustments and Submit Final Program: (35 min)

1. Students will test their prototypes
2. They will make adjustments as needed to improve performance and debug their code
3. Students will submit their final programs as their final submission

Summarize and Conclude: (10 min)

Discuss different ways students overcame challenges during this phase and how these solutions can give insight into real-life solutions.

Assessment: Summative and Formative

Summative assessments can be made during the testing and improvement phases of each lesson. Observing how students overcome obstacles and challenges as they arrive at their final programs.

A formative assessments can be done at the end of the unit. What types of sensors were used, what device settings should be used for a given maneuver, what are the similarities and differences between the autonomous vehicles in the real world vs. the robots in the classroom, etc.

Final Project: Program a robot to successfully all three phases of self-driving cars will be part of the formative assessment.