Lesson Title	Automated Vehicle Programming Design (Note: This lesson plan was designed for a robotics course)
Length of Lesson	
Created By	Dustin Spayde
Subject	Robotics
Grade Level	11-12
State Standards	
DOK Level	DOK 4
DOK Application	Design, Create, Apply Concepts, Analyze, Critique, Connect
National Standards	9-12: B (physical); E (technology)
Graduate Research Element	Developing Automated Systems

Student Learning Goal:
Physics: 1. Apply inquiry-based and problem-solving processes and skills to scientific investigations: (a) Use current technologies such as CD-ROM, DVD, Internet, and on-line data search to explore current research related to a specific topic; (g) Collect, analyze, and draw conclusions from data to create a formal presentation using available technology (e.g., computers, calculators, SmartBoard, CBL's, etc.)
4. Discuss the characteristics and properties of light and sound: (a) Describe and model characteristics and properties of mechanical waves; (e) Investigate and draw conclusions about the characteristics and properties of electromagnetic waves.
6. Analyze and explain concepts of nuclear physics: (b) Defend the wave-particle duality model of light, using observational evidence. Quantum energy and emission spectra.

For student to practice authentic scientific processes in the field to collect data; to analyze the data collected; and to create a product to reflect their understanding of the physics standards that focus on atomic binding energies and high energy emission spectra.

National Science Education Standards of Content 9-12
A (Inquiry): identify questions and concepts that guide scientific investigations.
E (Science and Technology): abilities of technological design; understanding about science and technology

Materials Needed (supplies, hand-outs, resources): A Lego Mindstoms NXT kit per 2-3 groups of students, Access to computers (one for each group) with USB ports and the Lego Mindstorms NXT software installed on each, multiple measuring tapes/rulers, a small poster board with a large 360 degree protractor drawn in the center (to determine how the robots turns). Basic materials to create a simple obstacle course that the vehicle will navigate. (I suggest using colored paper, empty cereal boxes, and tape. A sample obstacle course layout and building instructions are provided in the notes section.)

Lesson Performance Task/Assessment:
Each team will present their algorithm to be critiqued by the class. Students will learn basic programming logic, the basics of working in a design team, presentation skills, and critiquing fellow students.

Lesson Relevance to Performance Task and Students:
An automated vehicle such as this could easy be found in many factories and ports around the world. Developing its navigation system is an applicable task for a many engineering fields.

Anticipatory Set/Capture Interest:
As the class enters the room, the course and robots should already be laid out. Propose that a factory is looking for an automated vehicle to deliver parts around the factory. Explain that the students will be split into design groups and that they will be competing for the factory's contract. The team with the fastest completion time will win the contract. In the case of a tie, the team whose program takes up less memory will serve as the tie breaker. It may worthwhile to award the winning team some incentive as the "contract."

Guided Practice:
Day One: The task, rules, and resources are laid out before the students. Student teams are determined by the instructor. The teacher should also demonstrate the obstacle course being correctly run (a simple program is included for the provided obstacle course).
Rules:

- Students may measure the obstacle course with the provided measuring tape
- Students may use the protractor poster to measure how long the motors must run to complete the proper angle for a turn.
- Students may use the measuring tape to gage the distance the robot moves pre motor rotation.
- (Optional) Students are not allowed to test run within the actual obstacle course.
- Teams have two days to program. (This is assuming that the students have already been briefly introduced to the software. If this is not the case a longer design period will be required.)
- The team with the fastest run wins the contract.
- In the event of a tie, the team with the smallest (in terms of memory) program wins.

Day Three: Each team will run the obstacle course once. Each run will be timed by the instructor or a third party. The results will be placed on the board after each run. And a winning team will be determined. If time allows, teams will begin design reviews, a presentation explaining their algorithm to the rest of the class. Students should inquire and critique the presenting team's design. (Some important questions to ask during these design reviews are; "What are the limitations of your design?", "What would happen if the layout of the factory changed?", "Would your robot still function properly without going off course? Why or why not?", "How could you improve your design?")

Day Four: Complete the algorithm presentations/critique sessions.

Independent Practice:
Students will be presented with the obstacle course and the prebuilt automated vehicles. Students are then split into design teams. Each team is tasked with designing a program that allows the vehicle to complete the course in the shortest amount of time. Multiple teams will share one robot for testing purposes. Teams will then compete for the shortest to complete the course.

Day One and Two: Teams will design their programs using the available resources.

Day Three: Students will upload their programs and record the memory it takes up on the unit.

Day Four: Students will explain their algorithm to the class in a design review format.

Remediation and/or Enrichment:
Remediation: individual IEP; partner help throughout lesson; shorten parts of assignment; focus upon smaller elements of the process

Enrichment/Extension:

Set up a seminar which includes students from other science classrooms as an audience for the student team design presentations.

Check(s) for Understanding:

Day One: Have all teams designed for at least one obstacle?

Day Two: Have all teams completed their designs to navigate the course?

Day Three: Do the students understand why their robot preformed as it did?

Day Four: Can the students effectively explain their algorithms?

Overall: “What are the limitations of your design?”, “What would happen if the layout of the factory changed?”, “Would your robot still function properly without going off course? Why or why not?”, “How could you improve your design?”

Closure:

Day One: Allow the students to work as long as possible, while the instructor browses each group’s work to clarify any misconceptions. Finish by informing the students to save their work a being prepared to finish their programming during the next class period. Encourage them to design for the next period outside of the classroom.

Day Two: Allow the students to work as long as possible, while the instructor browses each group’s work to clarify any misconceptions. Finish with students saving their work.

Day Three: Discuss the applications of automated vehicles

Day Four: Close with an investigation into why the winning design actually won.

Possible Alternate Subject Integrations:

*Math – can manipulate mathematical expressions to isolate needed variables

*Language Arts – needs to deliver a public presentation of an engineering design

*Programming – Basic logic and algorithm models

Teacher Notes:
Include Sample Obstacle course, Building instructions, & Robot program.