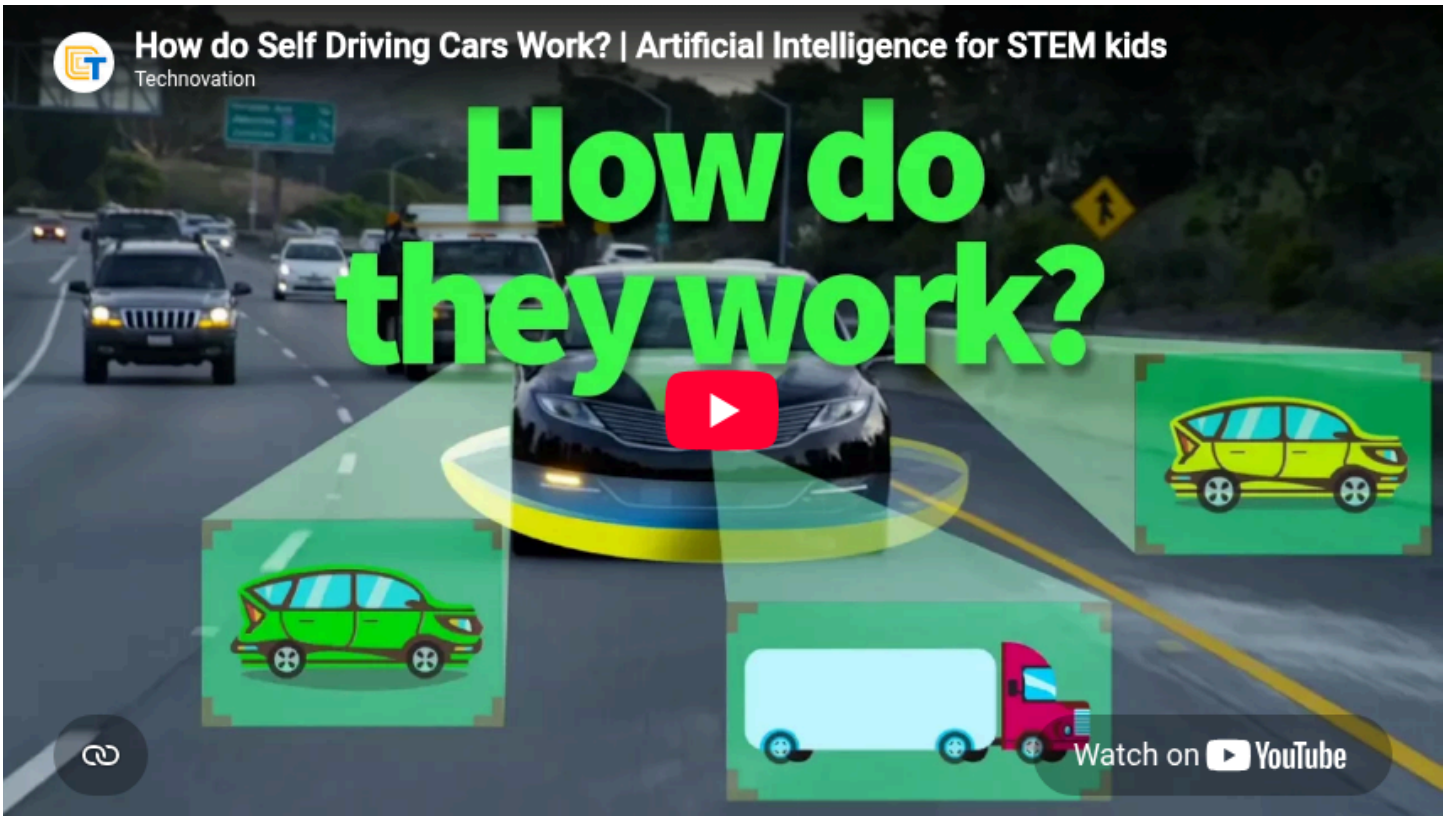
AI & Transportation



Lesson Plan: AI in Transportation – Computational Thinking, Decompos

Objective:

- Understand computational thinking concepts: decomposition and pattern recognition.

- Explore how AI helps with traffic management and obstacle avoidance to make driving safer.

Activities Brainstorm

Traffic Management and AI

Introduction to Computational Thinking:

Decomposition: Breaking down complex problems into smaller, manageable parts.

Pattern Recognition: Identifying patterns and trends to make predictions.

Activity 1: Exploring Traffic Management

-Discussion:

- What are the main problems in managing traffic (e.g., congestion, accidents)?

- How can breaking down these problems help us find solutions?

- How can recognizing patterns in traffic flow help improve traffic management?

Hands-On Activity: Traffic Simulation

Objective: Use decomposition and pattern recognition to understand and solve traffic problems.

- Instructions:

- Set up a simple city layout with roads, intersections, and traffic lights.

- Have students observe and note patterns in traffic flow (e.g., peak hours, congested areas).

- Break down the problem of traffic congestion into smaller parts (e.g., traffic light timing, car speed, lane usage).

- Discuss how AI can use these patterns to optimize traffic light timings and reduce congestion.

Group Discussion:

- How did decomposing the traffic problem help you understand it better?

- What patterns did you notice in the traffic flow?

- How do you think AI can use these patterns to manage traffic more effectively?

Part 2: Obstacle Avoidance and AI

Introduction to Obstacle Avoidance:

- Discuss the importance of obstacle avoidance in driving.

- Explain how AI can help vehicles detect and avoid obstacles.

Activity 2: Obstacle Avoidance Simulation with Sphero Indi

- Objective:Use computational thinking to program the Sphero Indi to navigate around obstacles.

- Instructions:

  - Set up a course with various obstacles.

  - Have students decompose the problem by identifying each obstacle and determining how to avoid it.

  - Program the Sphero Indi to detect and avoid obstacles using sensors.

  - Observe the Sphero Indi's movement and note any patterns in how it navigates the obstacles.

Group Discussion:

- How did decomposing the obstacle avoidance problem help you program the Sphero Indi?

- What patterns did you notice in the Sphero Indi's navigation?

- How can AI use these patterns to make driving safer?

Reflection and Closing Activity

Reflection:

- Discuss the key takeaways about computational thinking, decomposition, and pattern recognition.

- Have students share their thoughts on how AI can improve traffic management and obstacle avoidance.

Closing Activity:

- Draw or write about how you think AI can make driving and traffic management better in the future.

- Share and discuss your ideas with the class.

Materials Needed:

- Technology:Computers or tablets with internet access, Sphero Indi robots, Sphero Edu app.

- Supplies: Colored paper, small objects for obstacles, city layout for traffic simulation.

- Other: Access to videos or resources on traffic management and AI.

Integrating Standards

The next evolution of standards that would apply to this lesson, building on the Common Core and Next Generation Science Standards (NGSS), includes integrating the Computer Science Teachers Association (CSTA) standards and the International Society for Technology in Education (ISTE) standards. These standards emphasize computational thinking, digital citizenship, and the ethical implications of technology, which align well with lessons on AI, transportation, and bias. Here's how these standards can progress learning from previous standards:

Common Core Standards:

- **Mathematics:** Applying mathematical practices such as problem-solving, reasoning, and modeling in the context of AI and transportation.

- **English Language Arts:** Developing skills in reading, writing, speaking, and listening through discussions, reflections, and presentations on AI and bias.

Next Generation Science Standards (NGSS):

- **Science and Engineering Practices:** Encouraging students to ask questions, define problems, develop and use models, and analyze and interpret data in exploring AI's role in transportation.

- **Crosscutting Concepts:** Understanding patterns, cause and effect, and systems thinking as they apply to traffic management and obstacle avoidance.

Computer Science Teachers Association (CSTA) Standards:

- **Computational Thinking (CT):**

  - **1A-AP-10:** Develop programs with sequences and simple loops, to explore traffic patterns and obstacle avoidance.

  - **1B-AP-11:** Decompose problems into smaller, manageable parts, such as breaking down traffic issues and obstacle scenarios.

  - **1B-AP-15:** Test and debug programs, relevant to programming the Sphero Indi for traffic and obstacle navigation.

- **Data and Analysis (DA):**

  - **1B-DA-06:** Organize and present data in a way that communicates patterns effectively, such as traffic flow and congestion data.

International Society for Technology in Education (ISTE) Standards:

- **Empowered Learner:**

  - **1.1.b:** Build networks and customize learning environments in ways that support the learning process, such as using Scratch and Sphero Indi for hands-on activities.

- **Digital Citizen:**

  - **1.2.b:** Demonstrate an understanding of the fundamental concepts of technology operations, such as how AI systems function in transportation.

- **Knowledge Constructor:**

  - **1.3.d:** Build knowledge by actively exploring real-world issues and problems, such as traffic management and obstacle avoidance using AI.

- **Innovative Designer:**

  - **1.4.a:** Use a variety of technologies within a design process to identify and solve problems, like improving AI systems for traffic and safety.

- **Computational Thinker:**

  - **1.5.b:** Collect data or identify relevant data sets, use digital tools to analyze them, and represent data visually to facilitate problem-solving and decision-making.

Math and Reading Standards Integration

Common Core State Standards (CCSS) for Mathematics:

1. **Mathematical Practices:**

   - **CCSS.MATH.PRACTICE.MP1:** Make sense of problems and persevere in solving them.

   - **CCSS.MATH.PRACTICE.MP4:** Model with mathematics.

   - **CCSS.MATH.PRACTICE.MP7:** Look for and make use of structure.

2. **Grade-Specific Standards:**

   - **CCSS.MATH.CONTENT.3.OA.D.8:** Solve two-step word problems using the four operations, representing these problems using equations with a letter standing for the

unknown quantity.

Two-Step Word Problems for Sphero Indi Activities

Problem 1: Navigating an Obstacle Course

Scenario:

You are programming your Sphero Indi to navigate an obstacle course. The course has two sections. In the first section, Indi has to move forward 5 blocks and then turn left to move another 3 blocks to avoid obstacles.

Problem:

How many blocks in total does Indi need to move in the first section of the course?

- Step 1: Calculate the total blocks moved forward.

- Step 2: Add the blocks moved after turning.

Equation:

Let x be the total number of blocks Indi moves.

$$x = 5 + 3$$

Answer:

$$x = 8$$

Problem 2: Avoiding Multiple Obstacles

Scenario:

In the second section of the obstacle course, Indi needs to move around two obstacles. First, it moves forward 6 blocks, then it turns right and moves forward another 4 blocks. After avoiding the first obstacle, it needs to turn left and move 2 more blocks to avoid the second obstacle.

Problem:

How many blocks does Indi need to move in the second section to avoid both obstacles?

- Step 1: Calculate the total blocks moved after the first obstacle.

- Step 2: Add the blocks moved after the second obstacle.

Equation:

Let \(y \) be the total number of blocks Indi moves.

\[y = (6 + 4) + 2 \]

Answer:

\[y = 10 + 2 = 12 \]

Problem 3: Calculating the Total Distance

Scenario:

After completing the first and second sections of the course, you need to calculate the total distance Indi has moved. In the first section, Indi moved 8 blocks. In the second section, Indi moved 12 blocks.

Problem:

What is the total number of blocks Indi moved to complete both sections of the course?

- Step 1: Find the total blocks moved in the first section.

- Step 2: Add the blocks moved in the second section.

Equation:

Let \(z \) be the total number of blocks Indi moves.

\[z = 8 + 12 \]

Answer:

\[z = 20 \]

Problem 4: Adjusting for Errors

Scenario:

During a test run, Indi miscalculated and moved 2 extra blocks in the first section. In the second section, Indi moved 3 blocks fewer than planned. Originally, Indi was supposed to move 8 blocks in the first section and 12 blocks in the second section.

Problem:

What is the adjusted total number of blocks Indi moved?

- Step 1: Adjust the blocks moved in the first section.

- Step 2: Adjust the blocks moved in the second section.

- Step 3: Add the adjusted blocks for both sections.

Equation:

Let w be the total number of blocks Indi moves.

$$w = (8 + 2) + (12 - 3)$$

Answer:

$$w = 10 + 9 = 19$$

Problem 5: Calculating Remaining Distance

Scenario:

Indi needs to move a total of 30 blocks to complete the course. So far, Indi has moved 20 blocks in the first two sections.

Problem:

How many more blocks does Indi need to move to complete the course?

- Step 1: Calculate the total blocks needed to complete the course.

- Step 2: Subtract the blocks already moved from the total.

Equation:

Let v be the number of blocks Indi still needs to move.

$$v = 30 - 20$$

Answer:

$$v = 10$$

- **CCSS.MATH.CONTENT.3.MD.B.3:** Draw a scaled picture graph and a scaled bar graph to represent a data set with several categories. Solve one- and two-step "how many more" and "how many less" problems using information presented in scaled bar graphs.

Common Core State Standards (CCSS) for English Language Arts:

1. **Reading Informational Text:**

   - **CCSS.ELA-LITERACY.RI.3.1:** Ask and answer questions to demonstrate understanding of a text, referring explicitly to the text as the basis for the answers.

   - **CCSS.ELA-LITERACY.RI.3.3:** Describe the relationship between a series of historical events, scientific ideas or concepts, or steps in technical procedures in a text, using language that pertains to time, sequence, and cause/effect.

2. **Writing:**

   - **CCSS.ELA-LITERACY.W.3.2:** Write informative/explanatory texts to examine a topic and convey ideas and information clearly.

   - **CCSS.ELA-LITERACY.W.3.7:** Conduct short research projects that build knowledge about a topic.

3. **Speaking and Listening:**

   - **CCSS.ELA-LITERACY.SL.3.1:** Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly.

   - **CCSS.ELA-LITERACY.SL.3.4: Report on a topic or text, tell a story, or recount an experience with appropriate facts and relevant, descriptive details, speaking clearly at an understandable pace.

Integrating These Standards into the Lesson:

Math Standards Integration:

1. **Traffic Simulation Activity:**

- **CCSS.MATH.CONTENT.3.MD.B.3:** Students can collect data on traffic flow and represent this data in scaled bar graphs or picture graphs. They can solve problems based on the data, such as determining which intersection has the most congestion and proposing solutions.

- **CCSS.MATH.CONTENT.3.OA.D.8:** As part of problem-solving, students can use addition and subtraction to calculate the number of cars passing through intersections at different times and use multiplication or division to predict future traffic patterns.

2. **Obstacle Avoidance Simulation:**

- **CCSS.MATH.PRACTICE.MP1 and MP4:** Students decompose the problem of navigating a course with obstacles and model their solutions using the Sphero Indi, focusing on understanding and solving the task step by step.

Reading Standards Integration:

1. **TED-Ed Self-Driving Car Dilemma:**

- **CCSS.ELA-LITERACY.RI.3.1 and RI.3.3:** Students watch the video and describe the sequence of events and ethical dilemmas presented. They can answer questions to demonstrate understanding and discuss cause and effect relationships in AI decision-making.

2. **Research and Writing:**

- **CCSS.ELA-LITERACY.W.3.2 and W.3.7:** After learning about AI in transportation, students can write informative texts explaining how AI helps manage traffic or avoid obstacles. They can conduct short research projects on different AI applications in transportation and present their findings.

3. **Discussion and Presentation:**

- **CCSS.ELA-LITERACY.SL.3.1 and SL.3.4:** Students engage in discussions about their observations and ideas on improving AI systems. They can report on their experiences programming the Sphero Indi and the patterns they observed, presenting their findings to the class.

Implementation Examples:

- **Traffic Simulation:**

  - Students collect and graph data on traffic flow, solving problems using the data.

  - Discuss how AI could use this data to optimize traffic light timings and reduce congestion.

- **Obstacle Avoidance:**

  - Decompose the task of navigating obstacles with the Sphero Indi and model the solution.

  - Discuss patterns in the Sphero Indi's navigation and how AI can use these patterns for safer driving.

- **Self-Driving Car Dilemma:**

  - Watch the TED-Ed video and discuss the sequence of events and ethical considerations.

  - Answer questions to demonstrate understanding and explore cause and effect in AI decision-making.

- **Research and Writing:**

  - Conduct short research projects on AI in transportation.

  - Write and present informative texts on how AI helps manage traffic or avoid obstacles.

This integration of math and reading standards ensures that students develop a well-rounded understanding of AI in transportation while building essential skills in problem-solving, data analysis, reading comprehension, writing, and presentation.

Based on the activities we've discussed so far, particularly in your lessons on AI in transportation and smart cities, the NGSS Standard MS-ETS1-1 could be applied as follows:

1. **Activity Involving Sphero Indi and Traffic Simulation**:

- During the lesson on traffic patterns and AI in transportation, students can use the Sphero Indi robots to simulate traffic scenarios.

- **Application**: Students develop models or simulations to observe traffic flow patterns, test different variables (such as speed limits or traffic light timings), and modify their designs based on the outcomes. This iterative process helps them understand how adjustments can optimize traffic efficiency, akin to city planning scenarios.

2. **Virtual Reality (VR) Tour of Smart Cities**:

- When exploring smart city initiatives, such as in Singapore or Tokyo, students can engage in virtual tours that showcase various technologies like smart traffic management and environmental monitoring.

- **Application**: While exploring these VR environments, students can analyze data and observe how these technologies work, then propose modifications or improvements to enhance efficiency or sustainability. This process aligns with developing models for iterative testing and design refinement.

3. **Scratch Programming for Recommendation Systems**:

- In the Scratch activity where students interact with a recommendation system, they can analyze patterns and biases in the system's recommendations.

- **Application**: Students can develop their own versions of the recommendation system, adjusting input parameters or algorithms to improve fairness and accuracy. This iterative testing and modification process aligns with the NGSS standard as they aim to achieve an optimal design.

Each of these activities involves creating models or simulations, testing them iteratively, and modifying them based on observed data or outcomes. This iterative design process not only helps students understand the concepts of AI and smart technologies but also reinforces engineering design principles outlined in the NGSS MS-ETS1-1 standard.

Coding: Writing instructions for a computer to follow.

Artificial Intelligence (AI): Technology that enables machines to learn and make decisions.

Robot: A machine capable of carrying out complex actions.

Algorithm: A set of step-by-step instructions to solve a problem.

Conditional Statements: If-then instructions in programming.

Cece and Max's Coding Adventure

Cece's digital eyes sparkled with excitement as she bounced into the classroom. Today was her favorite day of the week—**coding day**. Max, her best friend and coding buddy, was already at his desk, scribbling down notes.

"Hey, Max! Ready for another adventure in coding?" Cece asked, dropping her backpack and taking a seat.

Max grinned. "Always! I can't wait to see what Mrs. Smith has in store for us today."

Mrs. Smith, their favorite teacher, was standing at the front of the classroom with a big smile on her face. "Good morning, class! Today, we're going to learn about the basics of **coding** and how it's used in **artificial intelligence**."

Cece raised her hand. "Mrs. Smith, why is coding so important for AI?"

"Great question, Cece," Mrs. Smith said. "Coding is like giving instructions to a computer. When we write code, we're telling the computer what to do. In AI, coding helps us create programs that can learn, make decisions, and solve problems."

Cece nodded, understanding a bit more. Max leaned over and whispered, "I bet we're going to do something super cool today!"

Mrs. Smith clapped her hands. "Alright, everyone, let's start with a story. Imagine you are the creators of a brand-new **robot** that helps people in their daily lives. To make your robot

smart, you need to teach it how to do different tasks. And to do that, you'll need to write **code**."

Cece and Max's eyes widened. They loved stories, especially ones where they got to be inventors. Mrs. Smith continued, "Let's follow the adventure of Cece and Max as they create their own robot."

In a bright and bustling city, Cece and Max were known as the best young inventors. One sunny morning, they decided to build a robot named Botley to help people around the city. Botley was no ordinary robot; it could learn and adapt, thanks to the power of AI.

"First, we need to write some code to teach Botley how to move," Cece said, holding up her tablet. "We can use a programming language to give Botley clear instructions."

Max nodded. "Let's start with simple commands. Forward, backward, left, and right." He typed in the commands, and Botley began to move around their workshop.

Cece smiled. "This is fun! But what if we want Botley to do more complicated tasks, like sorting recyclables or delivering packages?"

Mrs. Smith paused the story and explained to the class, "When we write code for more complicated tasks, we need to use **algorithms**. An algorithm is a set of step-by-step instructions to solve a problem."

Back in their workshop, Cece and Max started working on an algorithm to teach Botley how to sort recyclables. Cece wrote down the steps:

1. **Identify** the object.

2. **Check** if it's paper, plastic, or metal.

3. **Place** it in the correct bin.

Max added the code to Botley's program, and soon, Botley was sorting recyclables like a pro. "Great job, Botley!" Max cheered.

Cece beamed. "Now, let's teach Botley how to deliver packages. We'll need to use **conditional statements**. These are like if-then instructions."

Mrs. Smith explained, "Conditional statements tell the robot to do something if a certain condition is met. For example, if the door is closed, then Botley should knock."

Cece and Max wrote the conditional statements for Botley. "If the address is correct, then deliver the package," Cece said as she typed. "If the address is wrong, then return to the workshop."

With their new code, Botley successfully delivered packages all over the city. The townspeople were amazed at how helpful and smart Botley was.

"Wow, coding really makes a difference!" Cece said, high-fiving Max.

"Definitely," Max agreed. "And it’s all thanks to learning how to code and understanding AI."

Mrs. Smith concluded the lesson, "See, class? Coding is the foundation of creating intelligent systems like AI. By understanding coding, you can create amazing things that can help people and solve problems."

Cece raised her hand again. "Mrs. Smith, can we write our own code today?"

"Absolutely, Cece! Let's get started," Mrs. Smith said, handing out tablets. "Today, you'll each get to write code for a simple AI program."

Cece and Max couldn't wait to dive into their next coding adventure, knowing that with each line of code they wrote, they were shaping the future.

Vocabulary Words:

1. **Coding**: Writing instructions for a computer to follow.

2. **Artificial Intelligence (AI)**: Technology that enables machines to learn and make decisions.

3. **Robot**: A machine capable of carrying out complex actions.

4. **Algorithm**: A set of step-by-step instructions to solve a problem.

5. **Conditional Statements**: If-then instructions in programming.

Cece and Max's story showed them that coding isn't just about computers; it's about creativity, problem-solving, and making the world a better place. And that was the best adventure of all.