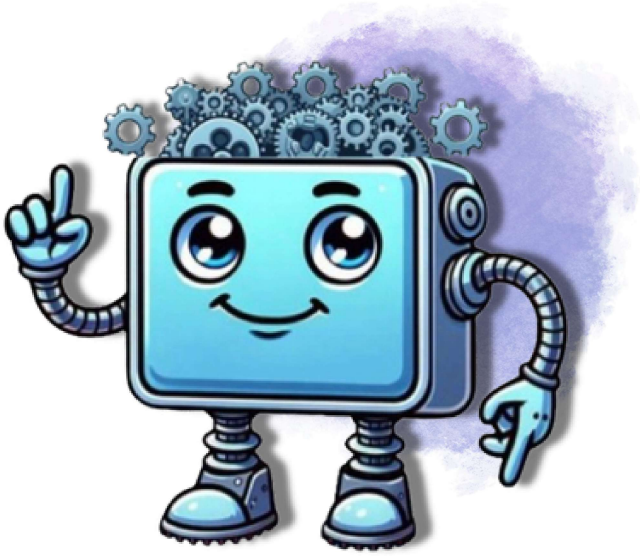
Name: Date:

Cece & The Engineering Design Process



Cece, Alex, and Priya were excited to start their summer engineering workshop. This wasn't a normal science camp—it was an innovation lab where middle schoolers got to act like real engineers.

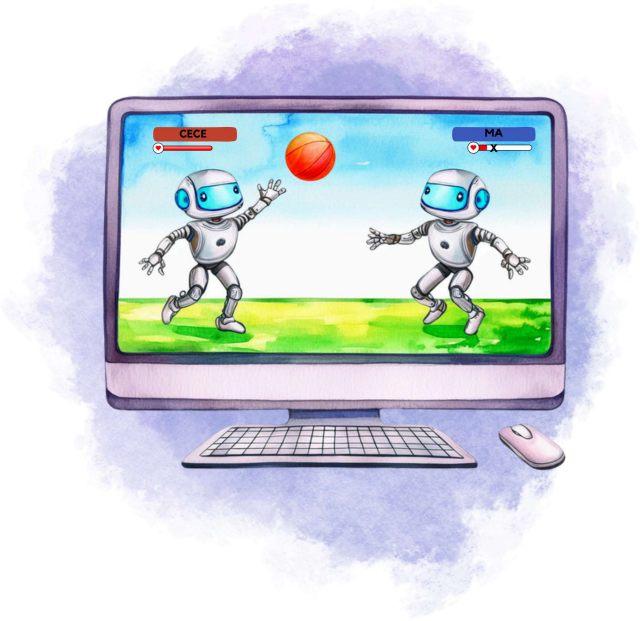
The city mayor had visited their school last month with a big problem: traffic lights in their town weren't smart enough. Sometimes cars sat waiting at empty intersections, wasting fuel and time. Other times, pedestrians had to wait too long to cross.

The challenge was clear: design a system that could use AI technology to make traffic lights smarter and safer. Their criteria were straightforward: Reduce waiting time for cars and people. Save energy. Keep the design simple enough for the city to test. But the constraints were tough: Limited budget. Only basic sensors and processors available.
Must work with the city's old power grid. Priya grinned. "This is exactly what engineers do—solve problems within real-world limits. Let's use the Engineering Design Process."

The group began to imagine solutions. Alex thought they could attach cameras to track cars. Cece suggested using simple pressure sensors in the road. Priya proposed training a small AI to predict traffic flow based on time of day.

"Remember, every design starts with a prototype," Cece said. "We'll need to build a model that can actually process data."
Alex pointed to the materials table. "We've got mini CPUs and GPUs, plus breadboards for circuits. If we wire it right, the function will be to switch the lights automatically."

"But GPUs use more energy," Priya warned. "That's a trade-off—faster AI, but higher power costs." They debated, sketching different structures on graph paper and running calculations. Efficiency was key, but so was accuracy.



After an hour, they created their first design: a circuit connected to a small processor that could switch LED lights based on car detection. The test didn't go well. The lights blinked randomly, sometimes switching too fast, sometimes not at all.

That's our feedback," Cece said. "Now comes iteration."
They adjusted the code, reorganized wires, and optimized the GPU to handle parallel image processing.

The second prototype worked better—it could recognize when a toy car rolled over the sensor and switch the light accordingly. "Not perfect," Priya admitted, "but we're getting closer."

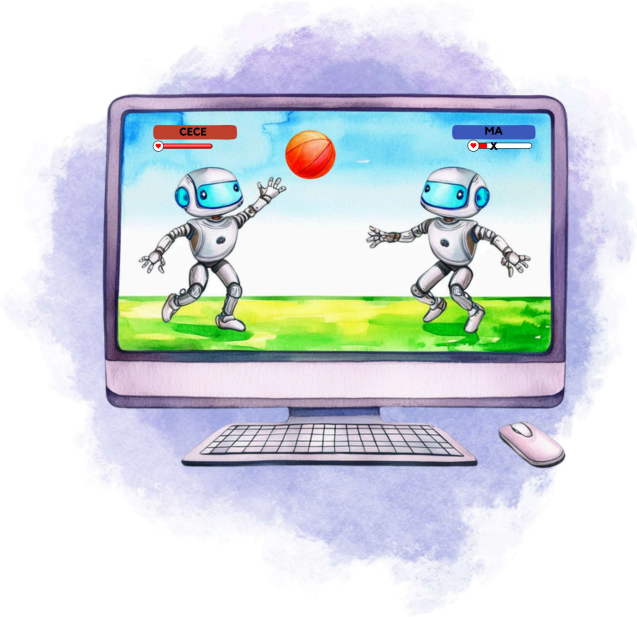
While testing, they discovered a constraint-bound solution: instead of fancy cameras, they could use low-cost infrared sensors. The system would predict traffic using a combination of real-time data and pre-programmed schedules.

"That's optimization," Alex said proudly. "We're balancing cost, performance, and energy."When they ran the test again, the model worked. Cars moved smoothly, pedestrians crossed safely, and the lights adjusted on their own.

At the final showcase, the mayor was impressed. "This isn't just a school project," she said. "This could save our city money and reduce emissions."The team realized that their small project connected to real AI engineering. Smart traffic lights, self-driving cars, and efficient power grids all used the same design process they had practiced.

Smart traffic lights, self-driving cars, and efficient power grids all used the same design process they had practiced. Cece summed it up: "Engineering isn't about having one perfect idea. It's about working together—collaboration—to test, improve, and keep pushing until the system really works."

The mayor loved the group's idea, but she wasn't the only one paying attention. A city engineer leaned in with a serious expression.



"Your model works in our workshop, but the real city system is much more complex. How will your design handle thousands of cars, emergency vehicles, or even power outages?"
The students paused. This was a tougher constraint than they had expected.

"If the power goes down," Alex said slowly, "our whole system might fail. That's dangerous."

Priya scribbled in her notebook. "What if we add a backup structure—a small battery pack? It would provide enough energy for the traffic lights to function for a few hours during an outage."

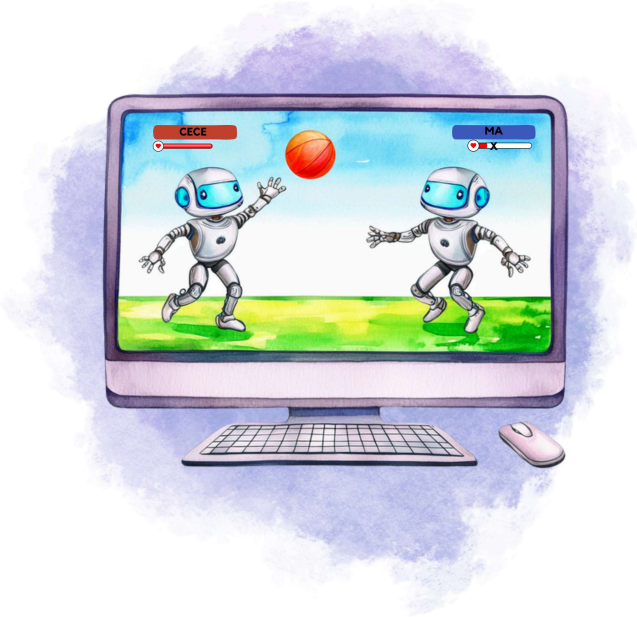
Cece nodded. "That could be our next iteration. It won't be easy, but engineers always prepare for unexpected conditions."

The team returned to their workstation, this time imagining how their design would function on a larger model of the city grid.
"We'll need more than just one prototype," Priya explained. "We'll have to test how the lights communicate with each other. That's what makes it a system instead of just separate devices."

Alex adjusted the wiring. "Communication requires sensors, but sensors increase the load on the processor. That slows it down."
"Classic trade-off," Cece said. "We need to balance speed with processing power."

They experimented with their GPU and CPU combination. The GPU could handle the fast recognition tasks—like detecting cars—while the CPU managed the schedule and communication between intersections.
"That's real optimization," Priya said proudly.

To prove their design, the students built a small cardboard model of downtown with toy cars and mini-LED traffic lights. They programmed their AI to predict when cars would appear and tested it with a randomized timer.



The first test worked… sort of. Cars flowed smoothly, but one corner of the model kept "jamming." Too many cars arrived at once, and the system didn't adjust quickly enough."That's our feedback," Cece reminded them. "The AI needs more training data."

They fed the system new scenarios: rush hour, school zones, even a simulated parade. Each time, the AI improved a little. Finally, after several iterations, the lights adjusted perfectly. Emergency vehicles got priority, pedestrians had safe crossings, and traffic stayed smooth.

When the team presented again, the mayor was impressed by their persistence.

"You turned your first idea into a functioning prototype, then kept refining it," she said. "That's what real engineers do." The city engineer added, "Your work shows real innovation. By combining affordable technology with smart planning, you've created something that's efficient, resilient, and practical."

The students beamed. They had solved a real-world problem through collaboration, careful planning, and plenty of testing.

As the workshop wrapped up, Cece leaned back in her chair.
"Do you think engineers working on self-driving cars or smart homes go through the same process we just did?" she asked.

"Absolutely," Priya replied. "They all follow the design process, balancing criteria and constraints to reach a constraint-bound solution."

Alex added, "And it's not just about building machines. It's about making technology that helps people—safer streets, faster commutes, less wasted energy. That's the true function of engineering."

The group packed up their cardboard city, knowing this was only the beginning. They hadn't just learned how to build a traffic light—they had discovered how engineers think, test, and improve the world around them.

Name: Date:

Glossary

1. An **Engineer** – A person who uses science and math to solve problems and build things.

2. **Civil Engineer** – An engineer who designs and builds things we use every day, like roads, bridges, and buildings.

3. **Electrical Engineer** – An engineer who works with electricity and technology to make machines, robots, and systems work.
4. **Program** – A set of coded directions that tells a robot or computer what to do.

5. **Sensor** – A tool on a machine that helps it "see" or "feel" things, like eyes or ears for robots.

6. **Data**: A collection of Information that a computer uses to learn.

7. **Training**: Then a computer practices using lots of examples to get better at a task.

8. **Roadway** – A path or street that cars, trucks, or robots can travel on.

9. **Obstacle** – Something in the way that must be avoided or passed through.

10. **Algorithm** – A list of steps or instructions a computer or robot follows.

Engineering Design Process (EDP) Steps

1. **Ask / Define the Problem**
   - What is the problem you are trying to solve?
   - Who will use your solution?
2. **Research / Explore**
   - Learn about the problem.
   - See how others have solved similar problems.
3. **Imagine / Brainstorm**
   - Come up with lots of ideas.
   - Don't worry about what's "best" yet — write down everything!
4. **Plan / Choose a Solution**
   - Pick the idea that seems the most realistic and effective.
   - Draw sketches or make a list of materials needed.
5. **Create / Build**
   - Make your design using your materials.
   - Follow your plan, but be ready to make adjustments.
6. **Test / Evaluate**
   - Try out your solution.
   - Does it work the way you want? What problems happen?
7. **Improve / Redesign**
   - Fix any problems or make it better.
   - Repeat the process as needed.