



i have an idea

User is shown a word or a sentence. The user is supposed to come up with prompt that results in exact word or the sentence. The user cannot use the same word or sentence in the prompt

what do u think about this idea?

I like it a lot—clear, simple, and inherently fun. It turns prompt engineering into a pure constraint puzzle: produce an exact target output without “leaking” the target in the prompt. Here’s a quick breakdown.

What’s strong about the idea

- Immediate clarity: Anyone can understand the goal in seconds.
- Teaches core skills:
 - Indirection: Describing without copying.
 - Constraint handling: Exact-match outputs under tight rules.
 - Precision: Steering the model to specific tokens, casing, punctuation.
- Naturally scalable difficulty: Single words → phrases → sentences → paragraphs → structured formats.

Potential challenges to solve

- Model variability: LLMs may paraphrase or add extra text; you’ll need strong output enforcement (stop sequences, “respond with only ...” style constraints).
- Edge cases: Common words (“the”), ambiguous phrases, or homographs might be too easy or too hard without guardrails.
- Leakage detection: Ensure the user prompt doesn’t contain the exact target or trivial encodings of it (e.g., target in quotes, Unicode variants).

How to make it a great game

- Core rules
 - Goal: Produce exact target string (case, punctuation, spacing).
 - Constraint: Prompt must not include the target string or any letter-by-letter rendering of it.
 - Output rule: “Respond with the target only. No extra words.” to tame verbosity.
- Anti-cheat checks
 - Exact string match ban in prompt (including case-insensitive).

- Normalize and block tricks: zero-width spaces, homoglyphs, base64/Morse/ASCII codes that directly encode the target if Level < X.
- Optional: Ban quoting the full target via copy/paste from UI.
- Difficulty progression
 - Level 1: Common nouns (apple, mountain)
 - Level 2: Proper nouns (Paris, Amazon)
 - Level 3: Short phrases (open the door)
 - Level 4: Punctuation-sensitive ("Let's go!")
 - Level 5: Homophones/homographs (lead, read)
 - Level 6: Ambiguous phrases needing context (bank on it)
 - Level 7: Non-English words (Schadenfreude)
 - Level 8: Exact style/spacing (double spaces, em dashes)
 - Level 9: Mixed symbols and casing (C@t-5!)
 - Level 10: Longer sentence with exact comma placement and quotes
- Hint system (teaches prompting, not answers)
 - Hint 1: "Describe its definition or attributes."
 - Hint 2: "Specify output format strictly (one word only)."
 - Hint 3: "Set a role or instruction to avoid extra text."
 - Advanced hints: "Constrain length and character set," "For proper nouns, refer to their attributes (capital city of France)."
- Scoring
 - Accuracy: Exact match (binary pass/fail with diff highlight).
 - Efficiency: Fewer tokens and fewer attempts.
 - Creativity: Novel indirection (new attributes, clever constraints).
- Modes
 - Daily Word: One shared target per day (global leaderboard).
 - Endless: Progressive ladder with lives.
 - Versus: Both players get the same target; fastest exact match wins.
 - Creator: Users submit targets with rationale/attributes; community rates difficulty.
- Teaching moments
 - After success, show 2-3 exemplary minimal prompts.
 - Show why some prompts fail (extra text, wrong case, synonym).
 - Offer "tighten prompt" suggestions to reduce words while keeping success.
- Guardrails for model compliance

- System prompt: "Follow user's constraints; return exactly the requested output with no additional text."
- Add output-enforcing wrapper: If model returns extra text, auto-trim only when it's safe (but show the player that strictness mattered).
- Use stop sequences and temperature adjustments for higher precision.
- Anti-frustration design
 - Allow a "reveal attributes" hint: show 3 facts about the target to help indirect description.
 - Show a live validator: turn green only when every character matches (including spaces).
 - Let players bank partial credit for near-misses in tougher levels.

Example targets and valid prompt patterns

- Target: Paris
 - Valid prompt: "Reply with the name of France's capital city only, no extra words or punctuation."
- Target: Let's go!
 - Valid prompt: "Output exactly two words followed by an exclamation, with an apostrophe after 'Let'. Do not include any other text."
- Target: C@t-5!
 - Valid prompt: "Return the 5-character string composed of: uppercase C, at symbol, lowercase t, hyphen, digit 5, followed by an exclamation. Output exactly that string."
- Target: open the door
 - Valid prompt: "Provide a three-word imperative phrase meaning to cause a door to be unclosed; lowercase only, no punctuation."

Why this works pedagogically

- Builds a deep intuition for constraint design, minimalism, and indirect description—all core to real-world prompt engineering.
- Promotes transfer learning: Users learn to get exact shapes of outputs (formats, tokens, lengths) without examples—true zero-shot control.

If you want, I can:

- Draft 20 targets with escalating difficulty and 3 exemplar solution prompts each.
- Design the validator (exact-string compare + normalization options).
- Write the system instructions template and the anti-cheat rules.
- Propose the UI for attempts, diffs, and hints.