Professor Sabine
CART 351
Nov 19, 2025

John Compuesto

PROJECT II - EXERCISE 4

Response 1 — Eamon: *Well of Fortune*

Eamon's *Well of Fortune* stood out as a playful reinterpretation of the classic fortune-telling experience, filtered through a clean and elegant interactive interface. One of the first things that caught my attention was the polish of the opening screen, with the subtle title animation, balanced typography, and the uncluttered menu layout, which immediately set the tone for the experience. It felt intentional, almost like a small narrative moment before the game even begins.

What I found most compelling was the mechanic where users choose whether they want a good or bad fortune and then actively fill in the blanks with their own words. It shifts the experience from passive consumption to co-creation. Instead of the program generating something entirely random, the user's language becomes part of the structure of the fortune itself. That makes every result feel personal and slightly unpredictable, which fits the theme of "fortune" very well.

From a design perspective, the project succeeds because the interaction is simple but conceptually clever, players don't need instructions to understand what to do, but once they engage, they start noticing how their choices affect the output. The code structure behind this, mapping user-submitted words into pre-written narrative templates, also suggests thoughtful planning.

If this project were expanded, I could imagine even more narrative branches or more stylistic variation depending on the fortune type. But in its current form, it already demonstrates a strong balance of aesthetic design and interactive storytelling.
Link: https://github.com/eamon1000000/cart_351_proj2_EamonFoley

Response 2 — Andre & Hugo: *Flaskemy Feud*

Andre and Hugo's *Flaskemy Feud* immediately reminded me of the energy of classroom quiz platforms like Kahoot, except this time the structure was unmistakably inspired by *Family Feud*. That combination of familiarity and reinterpretation made the experience feel both nostalgic and humorous. What surprised me the most was how closely the interaction follows the vibe of the original game despite being a much smaller project, the single-attempt answering, the survey-generated questions, and the pressure of choosing the right response all replicated that quick, competitive feeling.

The use of Flask on the backend was also an interesting technical choice because it mirrors how real-world quiz and polling apps operate, a server receives input, processes it, and updates the game logic. Even though the prototype does not yet support multiple rounds or ongoing scoring, the foundation is there for a more robust system. You can sense how easily this project could evolve into a full multiplayer experience with leaderboards or team-based play.

What I found most compelling was how they tied user-submitted survey answers into the question bank. That detail made the experience feel more collaborative, almost like the class itself contributed to shaping the game. With more time, I think expanding the number of prompts, refining the interface, and giving users more than one attempt could push the project into a fully polished game. But even as it is now, *Flaskemy Feud* does exactly what a Project II should do, test an idea, build a clear interaction loop, and make the user immediately understand the purpose of the experience.
Link:https://github.com/dreneder/cart-351-project_2/blob/main/CART-351-Project-II_Hugo_Andre/documentation/Andre_Neder-Project_2.pdf

Response 3 — Owen: *Sierpinski Triangle*

Owen's *Sierpinski Triangle* project was both visually striking and conceptually satisfying to interact with. Inspired by the Sierpiński carpet and related fractal logic, the project takes a mathematical concept and transforms it into an accessible, almost playful clicking experience. What surprised me most was the feeling of "building" a fractal one step at a time. Each click multiplies the triangles and deepens the pattern, letting users participate in the recursion rather than observing it passively.

The simplicity of the interaction, just clicking, helps make the project immediately understandable, even for users who may not know the math behind fractals. At the same time, the outcome is visually complex enough to feel rewarding. I especially appreciated how responsive the interface felt and how quickly the triangular subdivisions generated with each step.

One aspect that stood out was the current limit of about nine iterations. Even though the program eventually stops multiplying the triangles, this limitation actually reveals an interesting boundary between mathematical infinity and computational practicality. That tension made me think about how fractals behave in real code, at a certain point, performance, screen resolution, and rendering time all start to matter.

If Owen continues this project, a zoom-in or drag-to-navigate feature would be an exciting next step. It would let the user move deeper into the fractal, maintaining the illusion of infinite recursion. But even in its present form, the project succeeds at translating a mathematical idea into a hands-on interactive artwork.
Link: (couldnt find the link)