Snow day:
Three attempts at Tetris

15-113 – Monday 1/26

Quick stuff

- Welcome to my apartment. Stay warm, stay cozy. Office hours will also be remote today (see Ed for more details). Please keep your cameras on if possible.
- Congrats on finishing Project 1! We'll share info today on how to sign up for a presentation group
- This week's HW will involve making a functional game in Python. Details will be released tonight or tomorrow depending on how today's lecture goes
- Find anything interesting and AI-related? Any tips discovered while making your website? Please share on Ed!

Today's plan

- We'll attempt to create Tetris from scratch, several times, and each time we'll see how far we can get in 15 minutes.

- We won't easily be able to break off into groups over zoom, so chime in if you have questions, observations, or suggestions.

- Pay close attention so that you can replicate (and perhaps improve) any effective strategies on this week's HW!

Three approaches for today

- Simple prompt: "Make Tetris in Python with cmu_graphics."
- Plan, Adjust, Execute: "Create a detailed plan for writing Tetris in Python with cmu_graphics", we adjust the plan, AI executes the plan (also known as Plan-and-Solve)
- Detailed prompt with a full list of desired features and tests.

Which approach do you think will work best?

- Fill this out to let us know! https://bit.ly/4qGQLFe

Results

- Simple prompt: Well damn. With Sonnet 4.5, we reached working tetris in about 2:45 with reasonable checks for correctness.
- Plan Adjust Execute: Took 7 minutes, slightly more fully featured, but slower to get to a basic working state.
- (Semi)detailed prompt: Took almost 15, with one bug correction

- Note from office hours: Try Tetris + cmu_graphics with standard chatGPT! I think we didn't give Claude enough of a challenge, but chatGPT will demonstrate considerably more varied behavior on this task.

More prompting strategies

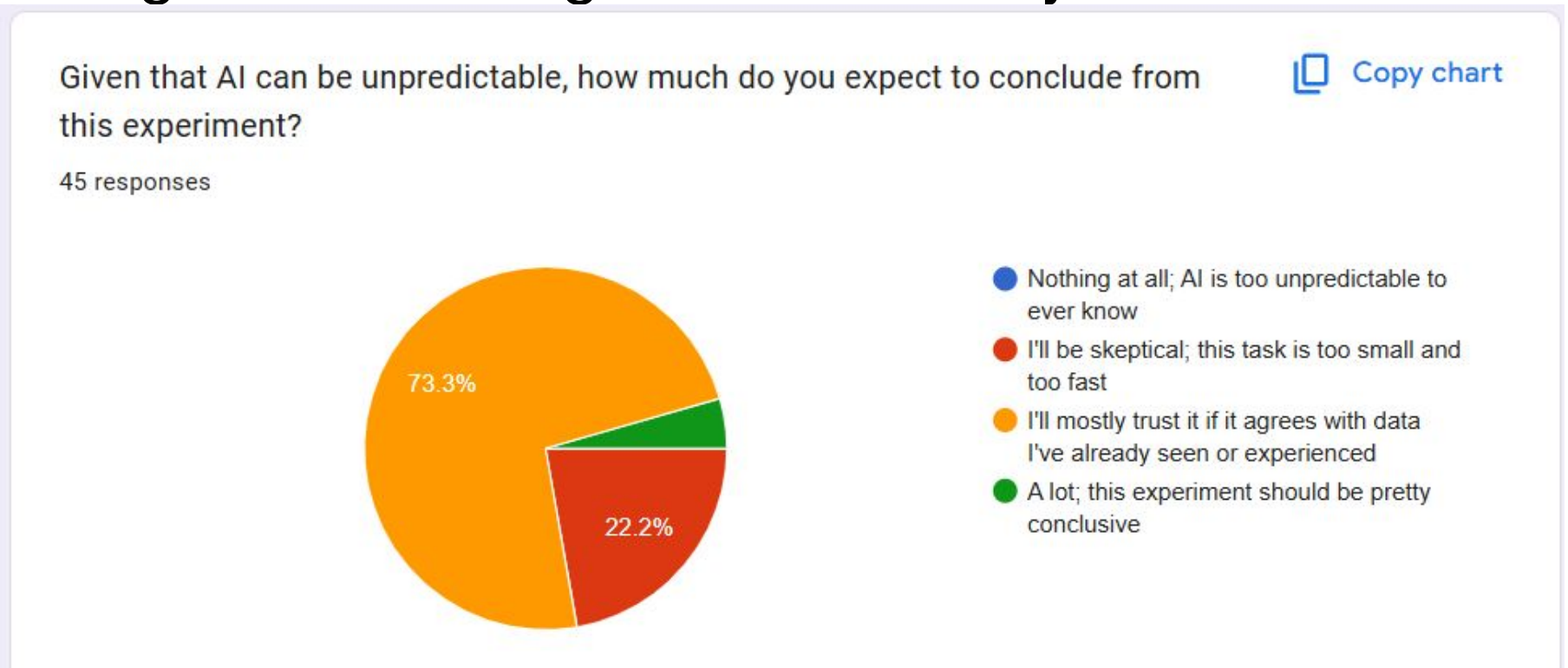
15-113 – Wednesday 1/28

Quick stuff

- HW2 is due Saturday 1/31 at 8pm
- Project 1 presentations Friday (be sure to sign up)
- No TA OH on Thurs/Fri (we'll be grading / listening to presentations) but Mike's Thurs OH is still on
- We're working our way towards bigger things: APIs, webapps, backend services, apps with AI feature integration (in other words, we're not going to linger on small games for long)

Monday debrief

- Monday's experiment was a little anticlimactic. Tetris is just not enough of a challenge for Claude anymore.



- Briefly, let's try just one prompt with ChatGPT

What *might* we still conclude?

- First, this is a moving target, but we should ask ourselves “Is this task rudimentary enough and well-defined enough for us to just use a simple but direct prompt?” If so, we might as well try that first.

- Using more complex prompting strategies *can* take more time.

- There *is* a limit to what a model can reliably nail in one quickly-written prompt. It *might* be roughly at Angry Birds (for Claude Sonnet 4.5, as of Monday).

Have you signed up for your presentation?

- Fill this out to let us know! https://bit.ly/3LYImxI

Let's have a quick peek at HW2

- HW2 is due Saturday 1/31 at 8pm, linked on schedule
- Core idea: Attempt to make as much progress as possible toward a clone of Crossy Road in **one hour.** You choose how you want to write your prompts, and what tools you'll use.
- Note: This one hour limit is *very important*. We can only compare our results if you strictly adhere to this. (The experiment is ruined if you spend 8 hours making a perfect clone and tell us it was done in 1 hour, especially if we believe you.)
- We'll consider what strategies were most effective for a given model (so we'll compare chatGPT to chatGPT, Claude to Claude, Gemini to Gemini)

Some common prompt suggestions:

- Avoid ambiguity. Be specific about what you want.
- Start with a high-level goal and then add details.
- Provide examples where relevant (e.g. “Data should be formatted like _____”)
- Offer additional information: “Feel free to ask any clarifying questions before you begin.”
- Ask for extra care: “Double-check that the code meets our initial specifications before presenting it to me, and iterate if necessary.”

Some common iteration suggestions:

- Provide feedback during iteration (e.g. “This is an improvement, but we still need _____.”)
- Add restrictions (e.g. “Make the game appear more visually modern, but do not change the underlying game logic.”)
- If iteration isn’t solving a bug, tell the AI to choose a different approach (e.g. “Do not hardcode piece rotations. Rotate algorithmically instead.” or “Rewrite the collision logic to ensure that collisions are correctly detected.”
- Ask for debugging output: “To help me test your code, print f‘collision ({object1}, {object2})’ whenever a collision is detected”

Many sources online (for code and in general)

- https://mitsloanedtech.mit.edu/ai/basics/effective-prompts/
- https://cobusgreyling.medium.com/12-prompt-engineering-techniques-644481c857aa
- https://ai.google.dev/gemini-api/docs/prompting-strategies
- https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/claude-4-best-practices
- https://medium.com/@the_manoj_desai/code-whispering-practical-prompting-techniques-that-make-llms-code-like-senior-devs-91f276d8c994

Magical Super-Prompts

- Lots of articles and reddit posts claim to have discovered the Very Best Do-Everything Prompt that will Make You Rich and Popular and Happy

- Mike's advice: Approach these with caution. Some may work well for specific tasks, but a one-size-fits-all approach will not likely fit *well*. Never discard your thoughtfulness.

- If you do find a prompt that works well, look at the *structure* and try to learn from that.

One last bit: Role instructions

- You can redefine how most models behave with a well-written role prompt. Several browser-based tools allow you to explicitly save and use these roles if you find a good one.

- Let’s try a few in chatGPT:
  - First, for fun, let’s have the AI role-play as a fictional character

  - Let’s try one that’s a bit more functional, and ask it to get to the point while avoiding empty phrases like “You are absolutely right and so smart for noticing that mistake.”

If time:

Don’t actually *do* HW2 in class, but just skim the assignment page, talk to your neighbors about how you might approach it, and maybe try just a few first prompts to see if they get you off to a good start.

(Later, once you’re ready to do the 1-hour part of the assignment, just be sure to start from scratch.)