

LLM Code Generation: Discussion

CS 485/698: AI-Assisted SE

“Greenfield” vs “Brownfield”

- A *greenfield* development project is one that lacks constraints imposed by prior work
 - Analogy: building on agricultural or park land

“Greenfield” vs “Brownfield”

- A **greenfield** development project is one that lacks constraints imposed by prior work
 - Analogy: building on agricultural or park land
- By contrast, a **brownfield** development project *is* constrained by prior work
 - Analogy: redeveloping a city block with existing infrastructure

“Greenfield” vs “Brownfield”

- A **greenfield** development project is one that lacks constraints imposed by prior work
 - Analogy: building on agricultural or park land
- By contrast, a **brownfield** development project *is* constrained by prior work
 - Analogy: redeveloping a city block with existing infrastructure
- In software engineering, most engineering work is brownfield: usually, you're hired to work on an existing codebase

“Greenfield” vs “Brownfield”

- A **greenfield** development project is one that lacks constraints imposed by prior work
 - Analogy: building on agricultural or park land
- By contrast, a **brownfield** development project *is* constrained by prior work
 - Analogy: redeveloping
- In software engineering, *most* projects are brownfield. Usually, you're hired to work on an existing system.

Last week's inclass activity was greenfield; Monday's was brownfield. Did you notice any differences in the AI's performance?

Hextris: show and tell

- How useful was the model's documentation for Hextris?
 - Did it help you understand the model's proposed changes later?

Hextris: show and tell

- How useful was the model's documentation for Hextris?
 - Did it help you understand the model's proposed changes later?
- Show of hands: how many people's models were able to add another color block on the first try?
 - For those of you who didn't raise your hands, what went wrong?

Hextris: show and tell

- Show of hands, who tried:
 - adding a “ghost block”

Hextris: show and tell

- Show of hands, who tried:
 - adding a “ghost block”
 - adding automated testing

Hextris: show and tell

- Show of hands, who tried:
 - adding a “ghost block”
 - adding automated testing
 - adding local multiplayer

In-class discussion: reflection essay questions

- Instructions:
 - Sort yourselves by question number (next slide)
 - Smaller numbers towards the front, higher numbers towards the back
 - Spend ~20 minutes discussing your essays and preparing to share with the class
 - Stand up and present for ~1-2 minutes
 - Expect questions or comments from me or your classmates

In-class discussion: reflection essay questions

- **Q1:** What kinds of programming tasks can LLMs do without (much) human supervision? What do these tasks have in common?
- **Q2:** Are there particular tools, technologies, or programming languages that LLMs are better or worse at generating code in or for? What do you think makes them well-suited/poorly-suited to LLMs?
- **Q3:** How detailed does the prompt need to be for "vibe coding" to be effective? Is a one-sentence description enough? How much is too much?
- **Q4:** Do you need to know anything about software engineering (e.g., any of the content of CS 490) to "vibe code"? Why or why not?
- **Q5:** How much code should be generated at a time when using LLMs for code generation? One function at a time? One file? Something else?
- **Q6:** How carefully do you need to review code that was generated by an LLM, compared to human-written code? How often can you use the generated code verbatim, versus having to modify it significantly?
- **Q7:** Are LLMs more likely than humans to write code with certain kinds of bugs? What kinds of bugs are LLMs more prone to? Do you trust humans or LLMs to write code with fewer bugs, and why?
- **Q8:** Is it easier or harder to debug LLM-written code than code written by another human? Why?
- **Q9:** How often do you find yourself having to reprompt the LLM with the same task in order to correct its mistakes?
- **Q10:** Is context summarization necessary when working on large projects with an LLM? Is it better to summarize the context or start a new session if the LLM "goes off the rails"?

Next steps

- Course project is starting soon, you should be thinking about possible projects
 - Goal: clone an existing web-delivered service, with at least one difference (e.g., different niche userbase)
- Project team formation by end of day Sunday (Feb 1)
 - Does the class want more time to make teams? We were supposed to have one more class by this point...
 - (Even if we delay team formation, P1 is still due at the same time: Feb 8.)