

Refactoring with AI

<https://devSteve.com/blog/posts/ai-refactor.html>

September 10, 2025

Introduction

Refactoring an old codebase used to be a nightmare. Thanks to AI tools like ChatGPT and Claude Code, that no longer seems to be the case. Using AI I was able to resurrect an old hobby project that hadn't been touched in over FIVE YEARS - an eternity in software development terms.

To be fair it took a few tries. The codebase uses some esoteric canvas animation code that Claude had a hard time updating. In the end I left the canvas animation code largely untouched and had Claude focus on things like modernizing the package versions and migrating the codebase from ES6 JavaScript to TypeScript.

If updating package versions and migrating to TypeScript don't sound like lofty enough goals, please keep in mind that prior to the refactor the code wouldn't even build and run. Plus, since the code wasn't originally written in TypeScript, there were initially *thousands* of TypeScript errors that needed to be resolved before the code would build. Literally thousands of errors, that's not an exaggeration. Updating all of those TypeScript errors by hand would've been an absolute nightmare.

In addition to the code changes handled by Claude, I relied on ChatGPT to help update some of the game's logos and graphics. Using AI for digital art seems to be a controversial topic. I find it no different than using AI for code or text, and I'm willing to debate anyone who cares to argue.

Project

[X's and O's](#) was a hobby project I started in early 2020 just after my dad passed away. Because of the timing and reason for its creation, the project has always meant more to me than the countless other hobby projects that have been abandoned over the years.

While my lack of artistic talent is clearly on display, the game has a few hidden gems to show that a significant amount of heart and soul was poured into the project. For example, after each play, there's a little matching cutscene.

In this clip we can see the "A" receiver caught the ball and got tackled in the 30-point zone (50 is the max you can score each play):

<Video redacted>

This is the cutscene that followed the play (it shows the "A" receiver with the ball getting stopped by the defense):

<Video redacted>

Behind the crappy graphics the game actually features some pretty advanced game mechanics. The players on offense run routes based on the play that was selected. The players on defense have their own playbook that they follow. At times it might look like a defender is doing something stupid, but it's usually just following the play call.

There's also a fair amount of math involved. When a pass is thrown, the game considers things like the receiver's distance from the ball, how fast the receiver is running, and in what direction the receiver is moving. A catch is not predetermined when the ball is thrown. Other players can interfere with the receiver and alter its route - usually resulting in an incompleteness or interception.

For the defense, math functions are used to determine the angle a defender should take when pursuing its target. If the code told a defender to run to its target's current location, then the defender would only be able to catch a still target. Instead, the defender must calculate the direction and speed of its target so it can choose a better path to intercept. Looking at the video above, we can see that the "L" defender didn't run directly toward the "A" receiver after A caught the ball.

Receivers without the ball turn into blockers, and more math is used so a blocker can try to get between the ball carrier and the closest defender. In the clip above we can see that the "B", "C", and "D" receivers all try to block for the "A" receiver - with "B" being the closest. The "X" linemen also run up and try to push the pile forward. The offense was incredibly close to scoring 50 points on the play, but the defense swarmed and stopped them just short of the line.

Goals

As mentioned in the introduction, as recently as a week ago (from the time of this writing) the X's and O's code would no longer build and run. This tends to happen to projects left untouched for many years. When initially trying to resurrect the project, the goals were pretty straightforward.

1. Update package versions and use modern React coding conventions.
2. Introduce TypeScript.
3. Get the code to build and run.
4. Update some of the site's graphics.

Claude Code was used for goals 1-3, and ChatGPT was used to help with the graphics. With the help of AI there may even come a day when the X's and O's game uses actual player characters and animations, rather than just a bunch of letters.

AI

Unfortunately I seem to have lost the exact prompts I used when using Claude Code to refactor the X's and O's codebase, but I'll do my best to paraphrase.

<Prompt redacted>

The simplicity of the prompt is really something. I've been told that Claude's internal context makes it unnecessary to assign a developer role (ex: an expert React and TypeScript developer) - and I used to laugh when people would say "please" to an LLM - but based on experience these practices don't seem to hurt the results, so I tend to keep using them.

Claude ran for several minutes and updated numerous files. When Claude finished I tried building the code but the build failed. Fortunately the build error told me the problem, which was that the code was using some old out-of-date version of Material UI. The next prompt went something like:

<Prompt redacted>

Sounds simple enough. Claude ran for several more minutes and managed to remove all references to the outdated @material-ui components. Claude also created a new set of UI components for things like buttons, checkboxes, and dialog boxes, and stored the new UI components in a src/components/ui folder. Makes sense to me, well done, Claude.

At this point the code still wouldn't build so it was on to the next prompt.

<Prompt redacted>

That was the last prompt I needed in order to get the X's and O's codebase to build from TypeScript. Not bad for a day's work. With the help of Claude Code, I was able to easily knock out goals 1-3.

1. ~~Update package versions and use modern React coding conventions.~~
2. ~~Introduce TypeScript.~~
3. ~~Get the code to build and run.~~
4. Update some of the site's graphics.

The X's and O's graphics obviously aren't great, but believe it or not, they're actually better now than they were when the project was shelved five years ago. In its first iteration, the X's and O's logo was literally just an X and an O.

<Image redacted>

I uploaded the original logo to ChatGPT and then gave it the prompt:

<Prompt redacted>

And the result? Pretty good if you ask me.

<Image redacted>

Next, I asked ChatGPT to help me generate some pixel art of fans in the stadium.

<Prompt redacted>

And this is what ChatGPT produced (again pretty darn good if you ask me):

<Image redacted>

The last graphics update I wanted to make before re-releasing X's and O's was to use a more realistic looking football. The original version of the game only used a brown ellipse with a white line. The current football graphics aren't perfect, but they're a huge improvement over a plain brown ellipse.

<Prompt redacted>

This last prompt was used multiple times to generate side-profile views of footballs with the laces at different positions. When the different images are looped together, we get a basic spiraling football animation.

<Animation redacted>

Conclusion

AI tools like Claude Code and ChatGPT not only make it possible to resurrect old hobby projects, they even make the process enjoyable. With the perspective of someone who has been writing code for more than twenty years (yikes), it seems clear that AI coding tools like Claude are the future (if not already the present).

The same seems to be true for digital art and media. I know it's an unpopular opinion, but I'll say it again, I don't see how using AI to generate art is any different than using AI to generate text or code. They're all the same. You either use AI, or you don't - and if you don't, it seems like an odd thing to virtue-signal about.