

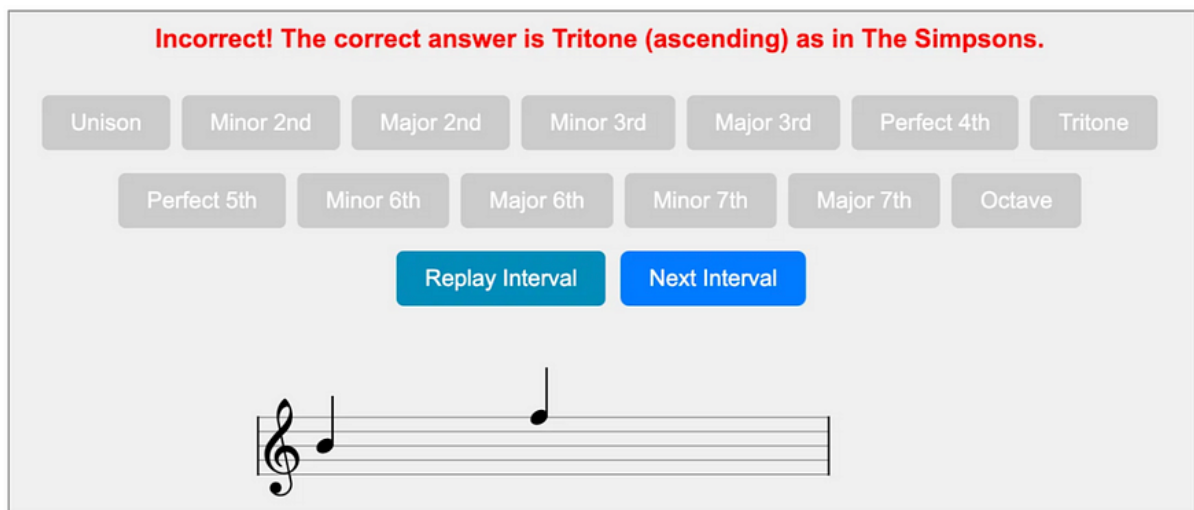
GPT-4 for Coding: 16 Observations and Tips

Author: Eric Silberstein

Claps: 85

Date: Mar 28

I asked GPT-4 to code an [app to help people learn musical intervals](#). The experience was mind-blowing. You can see the [full chat transcript](#). In an attempt to collect my thoughts, here are sixteen observations and tips for coding with GPT-4.



Screenshot of the app. Try it [here](#). Based on just a few instructions, GPT-4 coded the app including choosing The Simpsons theme song as an example of an ascending tritone.

Coding with GPT-4 is fun

You might think having GPT-4 do all the hard work would remove the puzzle solving that makes coding fun. But no, I found working with it as fun and addictive as *old-style* coding. You make a suggestion and seconds later it gives you the code for it, and then you see that suggestion come to life, and you get a nice, rewarding cycle going.

Coding is a killer application

Coding, and more generally computing, is a killer application for GPT-4. (I hope not in the sense that it will kill us.) I'm not sure why GPT-4 is so good at coding. Is it because there's so much information online about coding (e.g. github repos, blog posts, books, stack overflow) and all of that made its way into training data? Or is it because the type of *thinking* GPT-4 does is especially suited to coding? Or maybe it's equally good at tasks in other fields but I don't have the expertise to tell? But for sure this technology will change computing forever.

Like working with a pair programmer

It felt like pair programming with someone strong. Iâ€™ve done projects with engineers who have an almost superhuman ability to read code, write code, choose libraries, find bugs, use tools, and â€œplay computerâ€ in their head. It felt like working with someone like that, except this â€œpair programmerâ€ could read and write code in seconds.

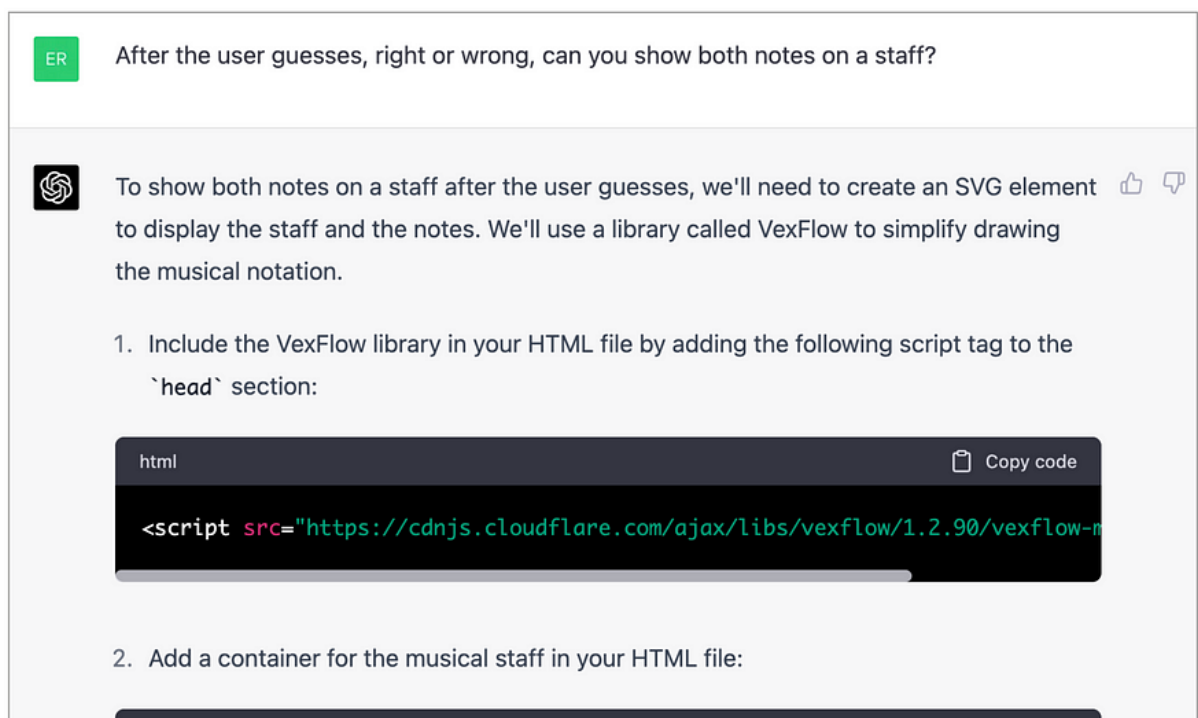
I was its hands and eyes, but thatâ€™s changing

My job, besides saying what I wanted and giving feedback, was to copy the code GPT-4 gave me into files, execute the code, test the app, and search the web when â€œweâ€ hit problems it couldnâ€™t solve on its own. However, with the [plugins](#) announced last week, still in private beta, ChatGPT should be able to run and edit its own code and do its own web browsing.

You donâ€™t need to be explicit or prescriptive

It has so much common sense and it knows so much about so much that you can give vague instructions or descriptions of problems and it will figure out the rest.


Hereâ€™s an example. All I did was ask it to show the notes on a staff. It understood what I was asking, picked a library, and wrote the code.



Hereâ€™s another example. Imagine even a year ago someone showing you these screenshots. You would never believe it was an AI on one side of the chat.

ER

When the user gets the interval wrong, I'd like the app to help them learn by mentioning a song that starts with that interval. For example, "Incorrect! The correct answer is Major 2nd as in 'mary had a little lamb'"



That's a great idea! You can add a list of songs for each interval and include the song information in the feedback message when the user gets the interval wrong. Here's an updated version of the `intervals` array and the `checkAnswer()` function:


javascript

Copy code

```
const intervals = [
  // ...
  { name: 'Major 2nd', semitones: 2, exampleSong: 'Mary Had a Little Lamb' }
  // ...
];
```

ER

Can you come up with the example songs? We'll need one for ascending and one for descending.



Sure, here's a list of example songs for each interval, both ascending and descending:

javascript

Copy code


```
const intervals = [
  { name: 'Unison', semitones: 0, ascendingExample: 'Twinkle, Twinkle, Little Star', descendingExample: 'Twinkle, Twinkle, Little Star' },
  { name: 'Minor 2nd', semitones: 1, ascendingExample: 'Jaws Theme', descendingExample: 'Jaws Theme' },
  { name: 'Major 2nd', semitones: 2, ascendingExample: 'Happy Birthday', descendingExample: 'Happy Birthday' },
  { name: 'Minor 3rd', semitones: 3, ascendingExample: 'Greensleeves', descendingExample: 'Greensleeves' },
  { name: 'Major 3rd', semitones: 4, ascendingExample: 'When the Saints Go Marching In', descendingExample: 'When the Saints Go Marching In' },
  // ...
];
```

Remind it of the current code

Even though it's telling you what code to write, it can't read the files where you're putting the code. In cases where it started giving me instructions that didn't line up with the code (for example, referencing a variable that wasn't there), I could fix that by reminding it of the code.

ER

Are you sure that's right? The current html has this: `<div id="staff-container" hidden></div>` (in other words already has a staff-container) and the js refers to it: `const staffContainer = document.getElementById('staff-container');`
`staffContainer.innerHTML = ''; // Clear previous staff`
`staffContainer.hidden = false;`



Apologies for the confusion, I didn't realize that you already had a staff container. In that case, you just need to update the CSS for the existing staff-container.

1. Add the following CSS to your stylesheet:

CSS

Copy code

```
#staff-container {  
  height: 200px; /* Adjust this value according to the height of your staff
```

It fixed its instructions once I gave it the current code. Staff-container wasn't something I added by hand, that was code it gave me earlier.


Browse the web for it

If you see it going in circles trying to fix a bug, you might think it means you need to fix the bug yourself. But I found I could get it to fix bugs by giving it new information.

ER

I think this which I found in a tutorial explains why it's not working: Notice that in the above example, even though we set the note names and durations correctly, we explicitly request the rendering of accidentals and dots.

This allows us to decouple rendering logic and notational semantics. For example, you would not want to render the # accidental on F# when the key signature already includes it (e.g. key of G).



You're correct. VexFlow requires us to explicitly add accidentals to the notes to be rendered. We can use the ``addAccidental`` method to achieve this. Let's modify the ``drawStaff`` function to add accidentals when necessary:

javascript

Copy code

```
function drawStaff(baseFrequency, interval) {  
  // ... (previous code)  
  
  const note1 = getNoteForFrequency(baseFrequency);  
  const note2 = getNoteForFrequency(baseFrequency * Math.pow(2, interval.sem
```


Example where I copied and pasted two paragraphs I found when googling for the problem and GPT-4 then corrected the code.

If you see a bug, let it solve it

My first instinct when I saw bugs was to solve them, but it was much faster to let it do it. Here's an example. It messed up the escaping of song titles (like not escaping the single quote in *I've Been Working on the Railroad*). As soon as I pointed out the problem, it fixed it. This may tell us something about how putting "attention" on the issue allows its "mind" to focus.

ER

That's great. But you have some stuff messed up in the quotes around the songs. You need to escape or use different quotes. Can you fix that?



I apologize for the oversight. Here's the corrected list of example songs with properly escaped quotes:

javascript

Copy code

```
const intervals = [
  { name: 'Unison', semitones: 0, ascendingExample: 'Twinkle, Twinkle, Little Star', descendingExample: 'Twinkle, Twinkle, Little Star' },
  { name: 'Minor 2nd', semitones: 1, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Major 2nd', semitones: 2, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Minor 3rd', semitones: 3, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Major 3rd', semitones: 4, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Perfect 4th', semitones: 5, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Tritone', semitones: 6, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Perfect 5th', semitones: 7, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Minor 6th', semitones: 8, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Major 6th', semitones: 9, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Minor 7th', semitones: 10, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Major 7th', semitones: 11, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' },
  { name: 'Octave', semitones: 12, ascendingExample: 'Joy Theme', descendingExample: 'Joy Theme' }
]
```

Here's a second example. I almost asked it to insert a pause between the sounds to see if that would solve it, but I'm glad I instead described the problem because it came up with a nicer solution.

ER

I'm hearing a click sound or something similar between the first and second pitch of the interval and after the end of the second pitch.



The click sound you are hearing may be due to the sudden stop of the audio when the notes end. To prevent the click sound, we can add a short fade-out at the end of each note using the `AudioParam.linearRampToValueAtTime()` method.

Update the `playNotes()` function in the JavaScript (`interval-trainer.js`) file as follows:

javascript


Copy code




```
function playNotes(baseFrequency, interval) {
  // ...
  // Add a short fade-out at the end of each note
  const audio = new AudioContext().createBufferSource();
  audio.frequency.setValueAtTime(baseFrequency, 0);
  audio.start(0);
  audio.linearRampToValueAtTime(0, 0.05);
  // ...
}
```

It's too eager to please and too confident

Watch out for this! It would rather bullshit an answer than tell you it doesn't know, and it doesn't express confidence levels like a human would. It seems to be biased toward providing solutions rather than telling you that something isn't possible. I saw this last week when it [hallucinated a google docs API that doesn't exist](#).

Here's an example from this app. I asked it to show a sample song when the user picks the wrong interval (see above). It chose the song *Watermelon Man* for the descending minor 7th, but I don't know that song. I asked for an alternative and it suggested a song with an *ascending* minor 7th. After some back and forth, I asked it to give me a list to choose from as shown here. Great! I know *The Boxer*, let's use that. The chorus is the "Lie-la-lie" part and the first two notes are *not* a minor 7th. But it was so confident! I'm pretty sure some if not all of the rest of these are wrong too. I guess there aren't many songs that start with a descending minor 7th interval and it doesn't want to admit that, even though it's certainly not at fault :)

 Could you list up to ten examples and then I'll pick one?

 Here are ten examples of songs with a descending Minor 7th interval:  

1. "There is a Light That Never Goes Out" by The Smiths (verse)
2. "Somewhere" from West Side Story (first two notes of the verse)
3. "I Just Don't Know What to Do with Myself" by Burt Bacharach (verse)
4. "The Winner Takes It All" by ABBA (first two notes of the chorus)
5. "So What" by Miles Davis (bass line)
6. "An American in Paris" by George Gershwin (main theme)
7. "Skating Away on the Thin Ice of the New Day" by Jethro Tull (verse)
8. "The Boxer" by Simon & Garfunkel (first two notes of the chorus)
9. "Kiss From A Rose" by Seal (first two notes of the pre-chorus)
10. "I Want to Hold Your Hand" by The Beatles (first two notes of the bridge)

Feel free to pick one that you think would be a good fit.

It gets pulled towards how things *should* be

Related to being too eager to please, I found that when it introduced inconsistencies, it seemed to be getting pulled toward how things should be or commonly are versus how they actually are.

In this example it gave me a code snippet with a variable named *intervalStats*. That was an appropriate name for the variable and an improvement on the original name it used, but the problem was it didn't tell me to change it in other places. When I pointed that out it reverted back to the old name.

```
score++,
intervalStats[correctIndex].correct++; // Update correct stats for the c
```

In this example, I asked it to generate a README file. It's incredible. However, the actual sound is not a "simple and clean piano sound" and you don't choose the interval from a dropdown menu.

Music Interval Ear Training App

This Music Interval Ear Training App is an interactive web-based tool designed to help musicians improve their interval recognition skills. The app randomly generates a pair of notes and plays the corresponding interval, challenging the user to identify the interval by ear. It also provides real-time feedback on the user's performance, allowing them to track their progress and identify areas that need more practice.

Features

- Random interval generation, including major, minor, perfect, augmented, and diminished intervals.
- Audio playback of intervals using a simple and clean piano sound.
- An interactive staff notation display that shows the generated interval, enabling users to visualize the notes as they listen.
- Real-time performance statistics, including the total number of attempts and the success rate for each interval type.
- A clean and user-friendly interface that is easy to navigate and understand.

How to Use

1. Open the app in a modern web browser.
2. Click the "Play" button to generate and play a random interval.
3. Use your ear training skills to identify the interval.
4. Select the interval you think you've heard from the dropdown menu and click "Submit."
5. The app will provide instant feedback on your response, displaying the correct interval on the staff notation and updating your performance statistics accordingly.

It will tell you how to debug

There's no way at present for it to run a debugger, add print statements, or even execute the code. (Although it sure feels like once it has access to a code execution environment it will be able to operate a debugger.) It will, though, tell you what to do!



Apologies for the confusion. This error is likely caused by the ``getNoteForFrequency`` function returning an incorrect note string or the note string not being recognized by VexFlow.



To find the root cause of this issue, I recommend adding some debugging to the ``getNoteForFrequency`` function by printing the input frequency, the calculated note string, and the octave. You can do this by adding ``console.log`` statements in the function:

```
javascript Copy code  
  
function getNoteForFrequency(frequency) {  
  // ...  
  
  console.log('Frequency:', frequency);  
  console.log('Note:', noteStrings[noteNumber]);  
  console.log('Octave:', octave);  
  
  return `${noteStrings[noteNumber]}/${octave}`;  
}
```

Then, inspect the browser console to see the output for the frequency, note, and octave when the error occurs. This will help you identify any potential issues in the ``getNoteForFrequency`` function or with the input values.

Itâ€™s great at naming things

Half of being a good programmer is coming up with good names for functions and variables. You can argue if itâ€™s half or not, but taking time to come up with good names forces clear thinking, encourages following conventions, makes the link between code and concepts in the outside world clear, and sets code up to be maintainable. GPT-4 comes up with good names.

JavaScript (interval-trainer.js):

javascript

Copy code

```
const instructions = document.getElementById('instructions');
const scoreElement = document.getElementById('score');
const feedback = document.getElementById('feedback');
const startButton = document.getElementById('start');
const intervalButtons = document.getElementById('interval-buttons');

const notes = ['C', 'C#', 'D', 'D#', 'E', 'F', 'F#', 'G', 'G#', 'A', 'A#', 'B', 'B#'];
const intervals = ['Unison', 'Minor 2nd', 'Major 2nd', 'Minor 3rd', 'Major 3rd', 'Minor 4th', 'Major 4th', 'Minor 5th', 'Major 5th', 'Minor 6th', 'Major 6th', 'Minor 7th', 'Major 7th', 'Octave'];

let score = 0;
let correctInterval;

function loadAudioFiles() {
  // Load audio files for each note
}

function generateRandomInterval() {
  // Generate a random interval (ascending or descending) and return it
}

function playNotes() {
  // Play the two notes based on the generated interval
}
```

GPT-4 came up with all the variable names, function names, and even the list of intervals with no help

Interval Trainer

Listen to the two notes and select the correct interval.

Correct: 0 | Incorrect: 0 | 0% correct

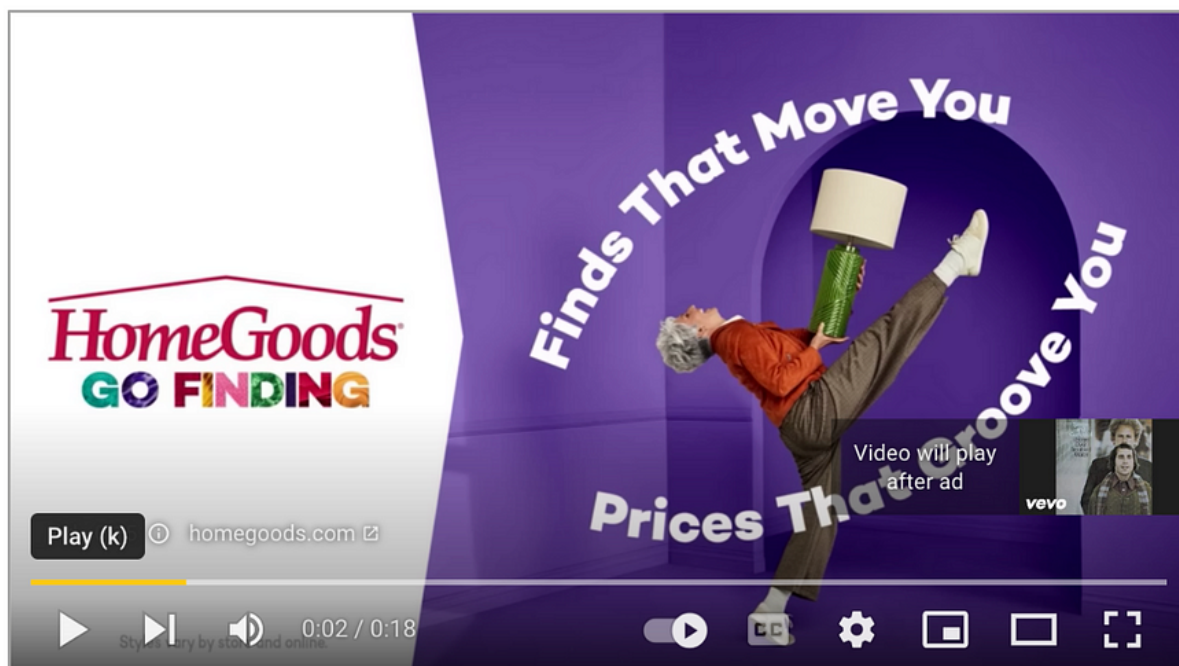
GPT-4 came up with the name "Interval Trainer" and all the other text on the screen. (If I'd asked for a more exciting app name I'm sure it would have come up with one.)

Other software starts to feel clunky

When interacting with GPT-4 / ChatGPT, you can be sloppy and incomplete just like when communicating with a human. It understands. Once you get used to that, other software feels clunky. For example, I was pasting my chat session into Medium, and being like *mon, you can't figure out what's text and what's code, determine the programming language for syntax highlighting, and format everything?*

Ads feel annoying

You get into a flow with GPT-4. I needed to break out of that flow and go to YouTube to figure out if the songs GPT-4 suggested had the descending minor seventh interval it claimed (see above). I found the ads at the start of YouTube more annoying than usual, I think because they broke the flow. I was moving so fast between GPT-4, VS Code, and the other browser tab, and GPT-4 was right there with me, and then I got stuck for twenty seconds waiting for an ad to play. (Maybe I should just subscribe to YouTube Premium!)



The 20 seconds of this ad felt like minutes

Browsing the web starts to feel tedious

One of the two issues I hit where I needed to go outside of GPT-4 was when sharps weren't being rendered on the staff (see above). To solve it I needed to do the typical thing of googling, looking at stack overflow, looking at tutorials, and looking at the docs for the music rendering library. I found this totally normal activity frustrating. It was hard to tell, as it often is, what information was credible and applied to my situation. The difference was this was now relative to working with GPT-4 where there was less hunting around. I also had the possibly wrong feeling that if GPT-4 could browse the web it would have figured it out faster than I did.

Do you even need to build an app?

In my few hours of fun I asked GPT-4 to build a standalone browser-based app. Anyone can use it. I even played with it on my T ride in this morning. However, you can see the day where using GPT-4 to generate this type of app will be unnecessary. A user will tell *future ChatGPT* to help them learn musical intervals. ChatGPT will then run training exercises within its own UI (voice or otherwise). It might choose to write "traditional" code to run the training exercises, but it will execute the code on its own and the user won't have to think about that. And of course it will be able to review the stats it collects during the exercises and use its LLM capabilities to give the user advice on how they're doing and where to focus.