

# Vibe Coding

*Über die Macht und Gefahr der Programmierung mit KI*

Willem L. Middelkoop

Apr. 15, 2025



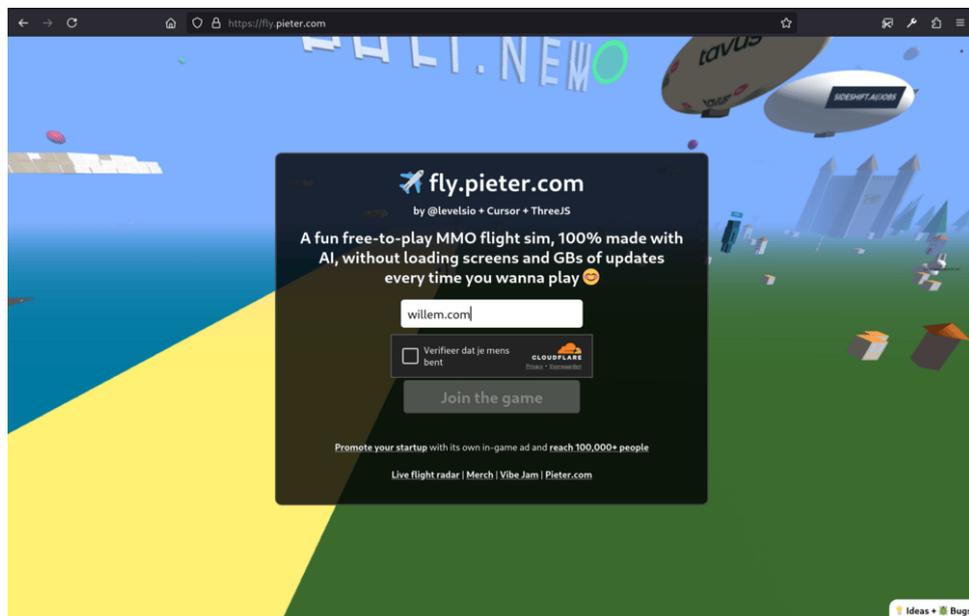
Mit über 48.000 \$ Umsatz pro Monat löste das Flugsimulatorspiel von Pieter Levels eine ganze Reihe von Innovationen aus. Ohne detaillierte Kenntnisse der 3D-Game-Engine-Technologie „vibe-codierte“ er sein Spiel mithilfe von KI. Kritiker seiner Arbeit wiesen auf Sicherheits- und Skalierbarkeitsprobleme hin, während Befürworter das erstaunliche Ergebnis lobten. Was können wir daraus lernen?

## Vibe Coding

Vibe Coding ist eine KI-gesteuerte Programmiermethode, bei der man ein Problem in wenigen Sätzen als Prompt für ein auf Programmierung abgestimmtes Sprachmodell beschreibt. Das Modell generiert Software und verlagert die Rolle des Programmierers auf die Anleitung, das Testen und das Verfeinern der Ausgabe. Befürworter behaupten, dass

es unerfahrenen Programmierern ermöglicht, Programme ohne umfangreiche Schulung oder Software-Engineering-Kenntnisse zu erstellen.

Pieter Levels hat das Spiel [fly.pieter.com](https://fly.pieter.com) entwickelt. In den sozialen Medien hielt er seine Bemühungen offen: von seiner ersten Idee bis zur Veröffentlichung. Er gewann Sponsoren für Werbung im Spiel und verkauft Premium-Upgrades (wie spezielle Flugzeuge) in seinem Spiel, wodurch er jeden Monat Tausende von Dollar einnimmt.

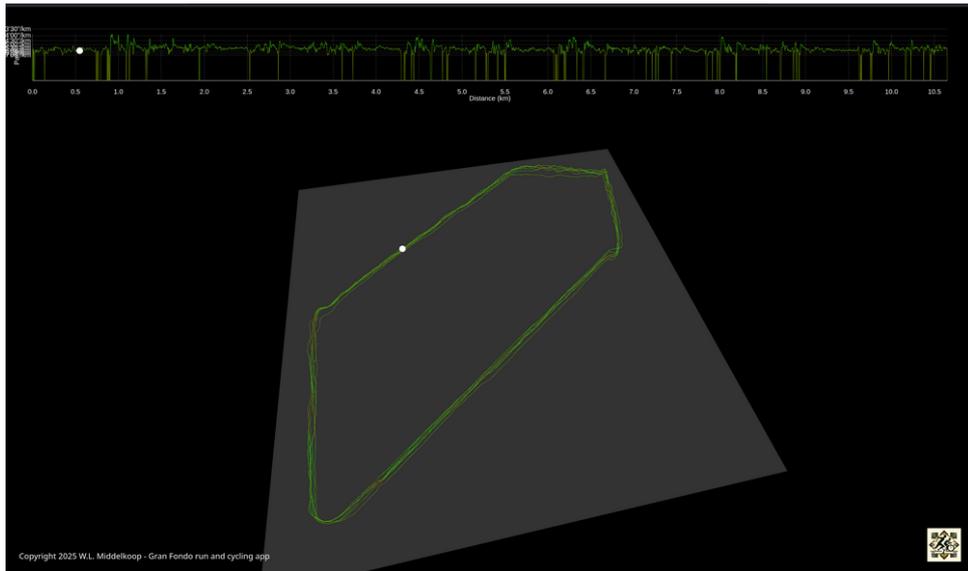


*KI-generierter Flugsimulator: fly.pieter.com*

Seine Erfolgsgeschichte warf Fragen über die Zukunft der Spieleprogrammierung oder sogar der Programmierung im Allgemeinen auf. Wenn ein Typ mit einem Laptop einen solchen Hit kreieren kann, wozu braucht man dann noch große Studios mit großem Budget? Oder umgekehrt gefragt: Was würde mit großen Studios passieren, wenn sie einen ähnlichen "Vibe-Coding"-Ansatz für die Produktion wählen würden?

## Selbst ausprobieren

Da ich selbst Erfahrung mit der Entwicklung von [a game](#) habe (Snake 97, über 40 Millionen Downloads), inspirierte mich Pieters Geschichte dazu, mich mit der 3D-Technologie zu befassen, die er für sein Spiel verwendet hat: [ThreeJS](#). Ein Spiel zu entwickeln ist schwer, es in 3D zu tun ist noch schwerer. Die Mathematik, die einer Welt in drei Dimensionen zugrunde liegt, stellt eine ganz neue Herausforderung dar. In vielerlei Hinsicht ist es anders als gewöhnliche Programme oder einfache zweidimensionale Spiele. Obwohl ich schon einige Experimente gemacht habe, fand ich die Lernkurve der 3D-Mathematik ziemlich steil. Hier kann die KI helfen.



*3D-Visualisierung von Trainingsdaten*

Für die [Gran Fondo app](#) habe ich die KI verwendet, um eine [drei dimensionale Präsentation einer aufgezeichneten Laufaktivität](#) zu generieren. Ich formulierte mein Ziel in einfachem Englisch und das KI-Modell generierte den Code, der benötigt wurde, um ThreeJS zu verwenden, um mein Training auf einer rotierenden Fläche anzuzeigen. Die ersten Ergebnisse waren nicht gut, aber ich verstand genug von den generierten Ergebnissen, um weiterhin nach spezifischen Verfeinerungen zu fragen. Einige Stunden später hatte ich etwas, das mir gefiel, aber vor allem hatte ich das Gefühl, viel über die Realisierbarkeit meines ursprünglichen Konzepts gelernt zu haben. Die KI ermöglichte es mir, schneller zu lernen.



*Spielkonzept-Artwork von 2013: HDL-Partikel im Blutkreislauf*

Ich beschloss, eine weitere Idee zu erforschen, die ich schon seit einiger Zeit aufgeschoben hatte: ein Spiel über schlechte und gute Cholesterinpartikel im Blutkreislauf eines Men-





*Produktivitätsparadoxon: Wenn die Bedienung der Maschine mehr Aufwand kostet als der Wert ihrer Leistung*

Ich habe die Entwicklung der Spiel-Engine mit KI nach einigen Tagen intensiver Experimente eingestellt. Ich erreichte einen Punkt, an dem ich mehr Mühe in die Verwaltung des KI-Modells steckte als in die eigentliche Spiel-Engine. Ich dachte, es wäre einfach besser, mir ein paar Bücher über die Technologien zu besorgen, sie zu lesen, selbst besser zu werden und dann die Spiel-Engine von Hand zu bauen. Wenn die Dinge komplex werden, scheinen unsere menschlichen Gehirne in der Lage zu sein, sich auf bestimmte Teile einer komplexen Aufgabe zu konzentrieren und gleichzeitig das größere Ganze zu respektieren.

## **Über die Torheit der "Programmierung in natürlicher Sprache"**

Zusätzlich zu den Grenzen des Kontexts, den ein KI-Modell verstehen kann, denke ich, dass es das Problem der Mehrdeutigkeit gibt. Ich empfehle Ihnen, die Worte von [Edsger Wybe Dijkstra](#) zu lesen, der in seinem [EWD667](#) über die Torheit der "Programmierung in natürlicher Sprache" nachdenkt. Eine eindeutige Beschreibung, frei von wirren Gedanken, bildet die Grundlage für zuverlässigen Code. Doch unsere Muttersprache lässt zu viel Raum für allerlei Unsinn und Fehler. Stattdessen sollten wir unsere Worte meistern, scharf denken und kühn spezifizieren.

Eine formale Sprache, wie eine Programmiersprache oder ein mathematisches Modell oder eine erschöpfende Ein- und Ausgabespezifikation, könnte ein erstaunlich effektives Werkzeug sein, um Mehrdeutigkeiten und Fehler auszuschließen. Einfach ausgedrückt: Sie erhalten die beste Ausgabe, wenn Sie die beste Eingabe liefern.

## Die KI mit Code auffordern

Um zu testen, ob die KI mir besseren Code liefern würde, beschloss ich, meine Sprache der Anfrage zu ändern. Anstatt zu beschreiben, was ich wollte, in natürlicher Sprache, lud ich ein Stück Code hoch, das ich zuvor geschrieben hatte, mit einigen sehr spezifischen Eigenschaften:

- der Code stammte aus einem Produktionssystem, ich wusste, dass er gut funktionierte
- der Code enthielt verschiedene Fehlerbehandlungsanweisungen, was zu seiner Robustheit beitrug
- der Code enthielt keine Syntaxfehler und verwendete logisch benannte Variablen und Funktionen
- der Code hatte einen sehr spezifischen Umfang, mit klar definierten Schnittstellen zur Außenwelt

Zusätzlich zu dem Code bat ich die KI ausdrücklich, einen bestimmten Aspekt des betreffenden Codes zu ändern. Sie musste die Verwendung eines bestimmten externen Programms (das NodeJS-Modul 'lwip') entfernen und durch ein anderes externes Programm (ImageMagick) ersetzen. Da sowohl [lwip](#) als auch [ImageMagick](#) Open-Source-Code sind, ist das KI-Modell mit diesen Programmen gut vertraut. Ich gab ihm das perfekte Rezept für seine Küche.

Der Code, den ich von der KI erhielt, war ein 99% perfekter Drop-in-Ersatz für meinen eigenen Code. Der generierte Code hatte einen Fehler, der dazu führte, dass einige Parameter nicht richtig escaped wurden, was zu einem Absturz führte. Ich konnte den Fehler manuell erkennen und von Hand beheben, da ich mich im Kontext dieser spezifischen Software voll und ganz auskannte. Nach einigen weiteren Tests beschloss ich, den KI-generierten Code in der Produktion zu verwenden, da die neue Bibliothek viel besser funktionierte als die, die sie ersetzte.

## Schlussfolgerung

Vibe Coding mit KI ermöglicht schnelles Prototyping und Lernen, wie der Erfolg von Pieter Levels zeigt, versagt aber bei komplexen Projekten, die Präzision erfordern. Es stellt sich die Frage: Erweitert die KI unser Potenzial oder beschränkt sie uns auf ihre Grenzen? Meisterschaft verbindet menschliches Denken mit der Kraft der KI. Denken Sie über Handwerkskunst nach: Wann schärfen Werkzeuge Ihre Fähigkeiten, und wann können sie sie stumpf machen?