

Codificación Vibe

Sobre el poder y el peligro de programar con IA

Willem L. Middelkoop

Apr. 15, 2025



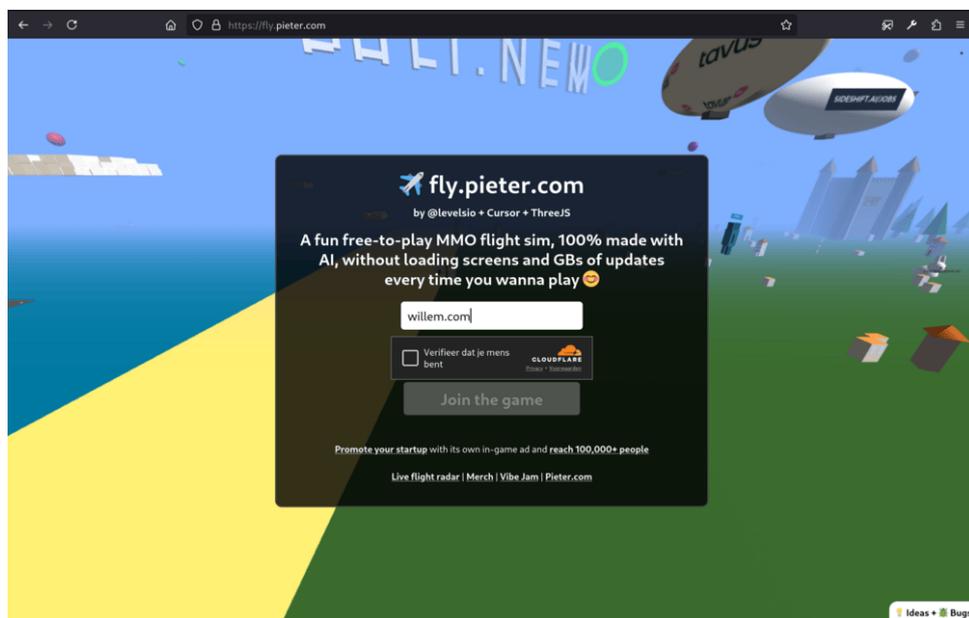
Facturando más de \$48 mil al mes, el juego de simulador de vuelo de Pieter Levels impulsó una oleada de innovación. Sin un conocimiento detallado de la tecnología de motores de juegos 3D, él 'codificó por vibra' su juego usando IA. Los críticos de su trabajo señalaron problemas de seguridad y escalabilidad, mientras que los defensores elogiaron el asombroso resultado. ¿Qué podemos aprender de esto?

Codificación por ambiente

La codificación por ambiente es un método de programación impulsado por IA donde describes un problema en unas pocas frases como una solicitud para un modelo de lenguaje ajustado para la codificación. El modelo genera software, cambiando el rol del programador a guiar, probar y refinar el resultado. Los defensores afirman que permite a los

programadores novatos crear programas sin una amplia capacitación o habilidades de ingeniería de software.

Pieter Levels creó el juego fly.pieter.com. En las redes sociales mantuvo un registro abierto de sus esfuerzos: desde su idea inicial hasta la publicación. Atrajo patrocinadores para anuncios dentro del juego y vende mejoras premium (como aviones especiales) en su juego, recaudando miles de dólares cada mes.

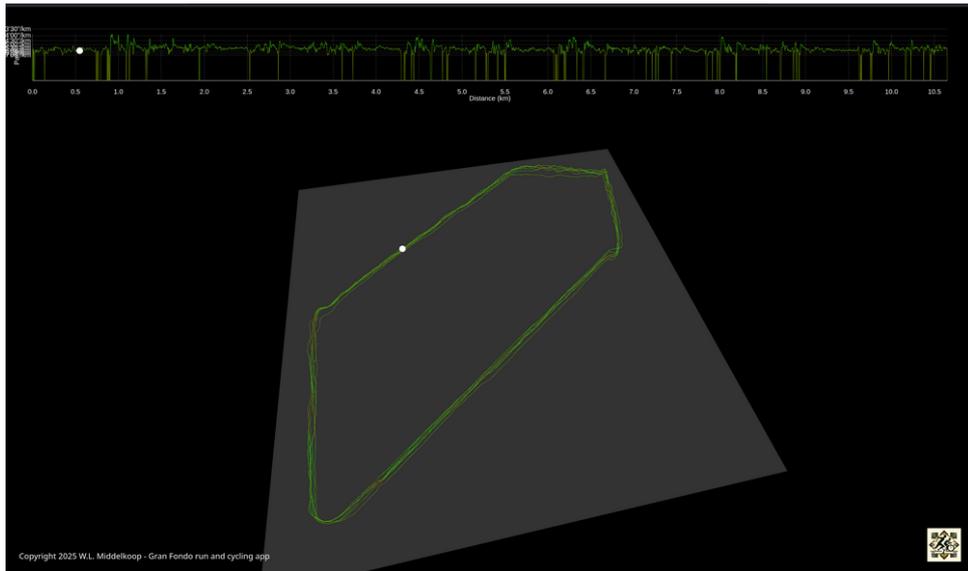


Simulador de vuelo generado por IA: fly.pieter.com

Su historia de éxito planteó preguntas sobre el futuro de la programación de juegos, o incluso de la programación en general. Si una persona con una computadora portátil puede crear un gran éxito como este, ¿cuál es la necesidad de estudios de gran presupuesto? O, preguntado inversamente, ¿qué pasaría con los estudios de gran presupuesto si adoptaran un enfoque de "codificación por ambiente" similar para la producción?

Probándolo yo mismo

Teniendo algo de experiencia en el desarrollo de [a game](#) yo mismo (Snake 97, más de 40 millones de descargas), la historia de Pieter me inspiró a investigar la tecnología 3D que usó para su juego: [ThreeJS](#). Construir un juego es difícil, hacerlo en 3D es aún más difícil. Las matemáticas que sustentan un mundo en tres dimensiones agregan un desafío adicional. En muchos sentidos, es diferente a los programas ordinarios o a los juegos bidimensionales simples. Aunque he hecho algunos experimentos antes, encontré la curva de aprendizaje de las matemáticas 3D bastante pronunciada. Aquí es donde la IA puede ayudar.



Visualización 3D de datos de entrenamiento

Para [Gran Fondo app](#) usé la IA para generar una [three dimensional presentation of a recorded running activity](#). Formulé mi objetivo en lenguaje sencillo y el modelo de IA generó el código necesario para usar ThreeJS para mostrar mi entrenamiento en un panel giratorio. Los resultados iniciales no fueron buenos, pero entendí lo suficiente de los resultados generados para continuar pidiendo refinamientos específicos. Algunas horas más tarde tenía algo que me gustaba, pero además sentía que había aprendido mucho sobre la viabilidad de mi concepto inicial. La IA me permitió aprender más rápido.

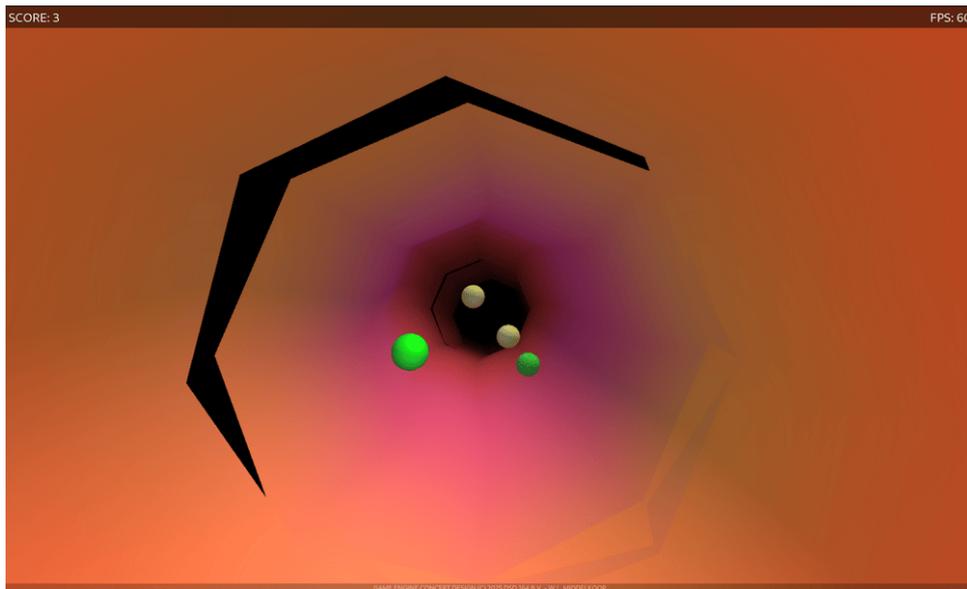


Arte conceptual del juego de 2013: partículas de HDL en el torrente sanguíneo

Decidí explorar otra idea que había dejado de lado por algún tiempo: un juego sobre partículas de colesterol buenas y malas en el torrente sanguíneo de una persona. Con aspectos de un [endless runner-style game](#), pensé que esto sería una buena opción para

seguir experimentando, ya que estos juegos cuentan con niveles generados por computadora. Construir un generador de niveles automatizado parece algo con lo que la IA podría ayudarme.

Aprovechando las nuevas capacidades 3D, pensé que el juego debería moverse dentro de las arterias en lugar del movimiento lateral común en los juegos 2D clásicos. Esto introduce desafíos como el recorte visual, gráficos distorsionados debido a que la posición de la cámara está fuera de la escena 3D. Otro desafío es la mayor complejidad de la detección de colisiones de los elementos del juego, un requisito para la mecánica del juego. El código generado por el modelo de IA creció y creció con cada iteración.



Motor de juego 3D en acción, observe los fallos gráficos en la pared de la arteria

Después de unas 800 líneas de código, comencé a experimentar las limitaciones del modelo de IA, ya que comenzó a cometer errores y fallos que había resuelto en iteraciones anteriores. Cada nueva iteración introducía nuevos problemas y el modelo seguía olvidando los refinamientos y las correcciones que había aplicado anteriormente. Esto se debe a los límites de las capacidades de contexto (tokens de entrada) que se importan para que los modelos de lenguaje grandes funcionen. En pocas palabras: no puede recordar lo suficiente como para continuar iterando sobre todo a la vez.



Paradoja de la productividad: cuando operar la máquina te cuesta más esfuerzo que el valor de su producción

Dejé de desarrollar el motor del juego usando IA después de algunos días de experimentación intensa. Llegué a un punto en el que estaba poniendo más esfuerzo en administrar el modelo de IA que en el motor del juego en sí. Pensé que simplemente sería mejor conseguir algunos libros sobre las tecnologías, leerlos, volverme más competente yo mismo y luego construir el motor del juego a mano. Cuando las cosas se vuelven complejas, nuestros cerebros humanos parecen ser capaces de concentrarse en partes específicas de una tarea compleja mientras respetan el conjunto más amplio tal como es.

Sobre la locura de la "Programación en lenguaje natural"

Además de los límites en el contexto que un modelo de IA puede comprender, creo que existe el problema de la ambigüedad. Recomiendo leer las palabras de [Edsger Wybe Dijkstra](#), reflexionando sobre la locura de la "programación en lenguaje natural" en su [EWD667](#). Una descripción inequívoca, libre de pensamientos confusos, forma la base del código confiable. Sin embargo, nuestra lengua materna deja demasiado espacio para todo tipo de tonterías y errores. En su lugar, deberíamos dominar nuestras palabras, pensar con agudeza y especificar con audacia.

Un lenguaje formal, como un lenguaje de programación o un modelo matemático o una especificación exhaustiva de entrada y salida, podría ser una herramienta increíblemente eficaz para descartar la ambigüedad y los errores. En pocas palabras: obtendrás el mejor resultado cuando proporciones la mejor entrada.

Solicitando a la IA con código

Para probar si la IA me daría un mejor código, decidí cambiar mi lenguaje de solicitud. En lugar de describir lo que quería en lenguaje natural, subí un fragmento de código que

escribí anteriormente, con algunas características muy específicas:

- el código provenía de un sistema de producción, sabía que funcionaba bien
- el código contenía varias declaraciones de manejo de errores, lo que aumentaba su robustez
- el código no contenía errores de sintaxis y utilizaba variables y funciones con nombres lógicos
- el código tenía un alcance muy específico, con interfaces claramente definidas con el mundo exterior

Además del código, le pedí específicamente a la IA que cambiara un aspecto particular del código en cuestión. Tenía que eliminar el uso de un determinado programa externo (el módulo NodeJS 'lwip') y reemplazarlo por otro programa externo (ImageMagick). Debido a que tanto [lwip](#) como [ImageMagick](#) son código abierto, el modelo de IA conoce bien estos programas. Le di la receta perfecta para su cocina.

El código que obtuve de la IA fue un reemplazo perfecto del 99% para mi propio código. El código generado tenía un error que causaba que algunos parámetros no se escaparan correctamente, lo que provocó un fallo. Pude detectar el error manualmente y corregirlo a mano, ya que me sentía completamente cómodo y conocedor dentro del contexto de este software específico. Después de algunas pruebas más, decidí poner en producción el código generado por la IA, ya que la nueva biblioteca funcionó mucho mejor que la que reemplazó.

Conclusión

La codificación por ambiente con IA permite la creación rápida de prototipos y el aprendizaje, como se ve en el éxito de Pieter Levels, pero falla con proyectos complejos que requieren precisión. Plantea una pregunta: ¿la IA amplía nuestro potencial o nos limita a sus limitaciones? La maestría combina el pensamiento humano con el poder de la IA. Reflexione sobre la artesanía: ¿cuándo las herramientas agudizan su habilidad y cuándo podrían desafilarse?