

Człowiek o AI (6)

* Analityk danych w firmie farmaceutycznej, autorka gry „Hexbreak”

Gra RPG (ang. *role-playing game*) – gra nastawiona na fabułę, rozwój postaci i podejmowanie decyzji wpływających na przebieg historii. Elementy zręcznościowe nie są najważniejsze, większy nacisk kładzie się na eksplorację świata, dialogi i wybory moralne. Kluczowe jest wcielenie się w bohatera oraz wczucie w jego rolę i opowiadanie historii, często w rozbudowanym, fikcyjnym uniwersum.

W prolog do gry „Hexbreak” można pograć tutaj:
<https://d209qoporefhg6.cloudfront.net/>



Najgorętszy język programowania? Angielski

Vibe coding i demokratyzacja tworzenia

Paulina WAWRO*

Najlepsze pomysły przychodzą mi w biegu. Dosłownie. Na leśnej ścieżce, gdzieś między korzeniami a zadyszka, myśli mają zwyczaj się rozplątywać. Wracalam w głowie do gry RPG, w którą ostatnio grałam, kiedy rozplątała się jedna szczególnie zuchwała: a gdyby postaci w takich grach napędzało AI? Gdyby po wspólnej przygodzie można było z nimi porozmawiać – naprawdę, jak z kimś żywym?

Pewnie wieki dzielą nas od czegoś takiego, pomyślałam. I natychmiast przyszła kolejna myśl, jeszcze zuchwalsza: a co, jakbym sama spróbowała to zrobić?

Więc spróbowałam. I tak powstał „Hexbreak” – moja gra RPG z autorską fabułą, muzyką, grafiką i bohaterami napędzanymi przez sztuczną inteligencję. Projekt, który jeszcze rok wcześniej wymagałby ekipy programistów, grafików i scenarzystów. Zrobiłam go sama, łącznie z silnikiem napisanym w Pythonie i Flutterze – języku, o którym wcześniej nawet nie słyszałam.

Jak to możliwe, zapytacie? Żeby to wyjaśnić, trzeba cofnąć się do pewnego posta, którego nikt wtedy nie wziął na poważnie.

„Najgorętszy nowy język programowania to angielski”

Dwudziestego czwartego stycznia 2023 roku Andrej Karpathy – współzałożyciel OpenAI, były szef AI w Tesli, jedna z najbardziej rozpoznawalnych postaci w świecie sztucznej inteligencji – wrzucił na X jedno zdanie. Brzmiało jak żart:

The hottest new programming language is English (Najgorętszym nowym językiem programowania jest angielski).

Dwa lata później już nikt się z tego nie śmieje.

W lutym 2025 roku ten sam Karpathy użył terminu, który słownik Collinsa ogłosił potem Słowem Roku: *vibe coding*. Opisał go jako „pełne poddanie się aurze, zaakceptowanie wykładniczego wzrostu i zapomnienie, że kod w ogóle istnieje”. Brzmi jak programista po psychodelikach? Otóż nie. To zupełnie poważna koncepcja. W praktyce wygląda tak: opisujesz zwykłym, ludzkim językiem, co chcesz osiągnąć, a sztuczna inteligencja zamienia to w działający program. Nie odpowiada, nie uzupełnia – pisze od zera. I to na poziomie, który jeszcze niedawno wymagał pracy zespołu deweloperów.

Dlaczego AI tak dobrze programuje?

Kod ma jedną cechę, która czyni go idealnym materiałem treningowym dla sztucznej inteligencji: można go zweryfikować. Uruchamiasz program i od razu wiesz, czy działa. Nie ma tu miejsca na interpretację ani subiektywność. Albo kod robi, co ma robić, albo nie. Kropka.

Ta właściwość umożliwiła przełom w uczeniu modeli językowych, znany jako RLVR – Reinforcement Learning with Verifiable Rewards, czyli uczenie ze wzmocnieniem z weryfikowalnymi nagrodami. Jak tłumaczy Karpathy: trenując modele na łamigłówkach matematycznych i programistycznych – gdzie maszynę można automatycznie sprawdzić – modele same zaczynają rozwijać strategie, które dla ludzi wyglądają jak rozumowanie. Uczą się rozkładać problemy na kroki.

Innymi słowy: AI przestaje tylko zgadywać najbardziej prawdopodobną odpowiedź – zaczyna uczyć się strategii prowadzących do poprawnego wyniku.

To zmiana fundamentalna. Wcześniejsze metody wymagały, żeby ludzie ręcznie oceniali, która odpowiedź „wygląda lepiej”. Nowe podejście nie pyta człowieka –

pyta kompilator albo kalkulator: „Czy to jest poprawne?” Jak ujął to jeden z badaczy: przeszliśmy od nagradzania za dobre wrażenie do nagradzania za weryfikowalną prawdę.

Dodajmy do tego miliardy linii kodu, na których te modele trenują – ponad osiemdziesiąt języków programowania, dokumentacja, zgłoszenia błędów, dyskusje techniczne – i mamy przepis na narzędzie, które rozumie nie tylko składnię, ale i kontekst.

Pod koniec 2025 roku, w ciągu zaledwie dwóch tygodni, trzej giganci technologiczni wypuścili swoje najmocniejsze modele do kodowania: Google Gemini 3 Pro, Claude Opus 4.5 od Anthropic i GPT-5.2 od OpenAI. To modele używane dziś w narzędziach takich, jak ChatGPT czy Copilot. Test wydajności SWE-bench Verified, który sprawdza, czy modele potrafią naprawić prawdziwe błędy w istniejących projektach, wszystkie trzy przekraczają teraz osiemdziesiąt procent skuteczności. Rok temu wyniki około siedemdziesięciu procent uchodziły za przełomowe. I żeby było jasne: te modele nie tylko piszą kod. One rozumieją istniejące projekty, poruszają się między plikami, wprowadzają poprawki i pilnują, żeby niczego po drodze nie zepsuć. Jeszcze niedawno taki poziom wymagał lat doświadczenia.

Liczby, w które trudno uwierzyć

Statystyki z 2025 roku jeszcze kilka lat temu brzmiałyby jak science fiction:

Według raportów branżowych czterdzieści jeden procent całego kodu tworzonego na świecie generuje lub wspomaga AI. Nie, to nie pomyłka. Prawie połowa. Osiemdziesiąt cztery procent programistów używa lub planuje używać narzędzi AI, a ponad połowa robi to codziennie.

W marcu 2025 roku Y Combinator – legendarny akcelerator startupów stojący za Airbnb, Dropboxem i Stripe’em – ujawnił statystykę, od której kręci się w głowie: co czwarty startup z zimowej kohorty 2025 ma bazę kodu w dziewięćdziesięciu pięciu procentach wygenerowaną przez sztuczną inteligencję. A Jared Friedman, partner zarządzający YC, dodał: „To nie tak, że finansujemy ludzi bez zaplecza technicznego. Każda z tych osób jest wysoce techniczna. Rok temu zbudowaliby produkt sami – ale teraz prawie cały kod pisze AI”.

Co to oznacza w praktyce? Wykwalifikowany programista potrafi teraz zrobić w weekend to, co jeszcze niedawno wymagało zespołu i miesięcy pracy.

Ale najciekawsze nie są same liczby. Najciekawsze jest to, kto z tej rewolucji korzysta.

Technologia do góry nogami

W kwietniu 2025 roku Karpathy opublikował esej pod tytułem „Power to the people”, w którym zauważył coś niespotykanego: duże modele językowe pokazują dramatyczne odwrócenie typowego

wzorca rozprzestrzeniania się technologii. Dają nieproporcjonalnie dużo zwykłym ludziom, podczas gdy korporacje i rządy mają trudność z szybkim wdrożeniem tej zmiany.

Chwila refleksji. W przeszłości przełomowe technologie trafiały najpierw do rządów i wielkich firm. Pierwszy komputer ENIAC – ważący dwadzieścia siedem ton – liczył trajektorie pocisków dla armii amerykańskiej. Internet zaczynał jako ARPANET: sieć zaprojektowana w latach sześćdziesiątych, żeby połączyć uniwersytety i instytucje wojskowe na wypadek ataku nuklearnego. GPS przez dekady pozostawał zarezerwowany dla wojska; cywile dostali do niego dostęp dopiero w latach osiemdziesiątych, i to w celowo okrojonej wersji. Zwykli ludzie zawsze stali na końcu kolejki.

Z dużymi modelami językowymi jest dokładnie odwrotnie.

ChatGPT stał się najszybciej rosnącą aplikacją konsumencką w historii – czterysta milionów aktywnych użytkowników tygodniowo. Bariera wejścia jest absurdalnie niska: modele są udostępniane tanio (często darmowo), są szybkie, dostępne w telefonie i mówią twoim językiem. Karpathy ujął to następująco: „Bill Gates rozmawia z GPT-4o dokładnie tak samo jak ty. Pieniądze nie kupią lepszego ChatGPT” (co teraz, ze względu na płatne subskrypcje, nie jest już do końca aktualne).

Sytuacja bez precedensu. Pojedynczy człowiek dostaje narzędzie, które zasadniczo zmienia jego możliwości – zanim korporacje zdążą się zorientować, jak to wdrożyć.

Karpathy pisze: „Ludzie mogą teraz vibe-codować aplikacje. O ile mi wiadomo, przeciętny człowiek nigdy nie doświadczył tak spektakularnego, tak szybkiego technologicznego skoku”.

W lipcu 2025 roku Lenny Rachitsky – autor popularnego branżowego biuletynu – zapytał swoich czytelników: co zbudowaliście? W ciągu doby dostał ponad tysiąc odpowiedzi. . . I to jest moja ulubiona część tej historii.

Pewien ojciec kupił córce audiobooka, ale pliki nie były podzielone na rozdziały – a książka miała ich siedemdziesiąt. Napisał skrypt, który automatycznie transkrybuje nagranie i tnie je we właściwych miejscach. „AI napisało kod praktycznie za pierwszym podejściem”. Ten pomysł zainspirował mnie samą do napisania skryptu, który przekształca niewygodne w użyciu audiobooki z YouTube na elegancko podzielone pliki mp3. Gotowe w mniej niż pół godziny.

Inny tata zbudował CarbScan – aplikację do szybkiego szacowania węglowodanów w posiłkach, żeby lepiej zarządzać poziomem glukozy u syna z cukrzycą. Używa jej codziennie.

CarbScan: <https://carbscan.ai/>

Pewien zapobiegliwy ojciec, czekając na narodziny dziecka, stworzył w dwa weekendy aplikację do monitorowania karmień, snu i pieluch.

Pewna mama zbudowała narzędzie, które zamienia emocje dziecka w spersonalizowaną bajkę na dobranoc.

Ktoś zrobił aplikację „Ile warstw ubrań dzisiaj?” – bo istniejące programy pogodowe miały za dużo funkcji. Ma teraz osiemdziesiąt pięć tysięcy użytkowników.

Ile warstw ubrań dzisiaj?: <https://layers.today/>

Dziennikarz New York Timesa zbudował LunchBox Buddy – robisz zdjęcie lodówki, a aplikacja proponuje, co włożyć synowi do pojemnika na drugie śniadanie. Gotowe w dziesięć minut.

LunchBox Buddy: <https://joyful-froyo-b4482b.netlify.app>

Ktoś inny stworzył wizualizator cięcia sklejk: wpisujesz wymiary deski, a program mówi, ile kawałków wyjdzie i ile resztek zostanie.

Mój faworyt? Rozszerzenie do przeglądarki, które pokazuje, ile osób właśnie czyta ten sam artykuł na Wikipedii co ty. Żebyś wiedział, że nie jesteś sam w swojej króliczej norze.

Każdy z tych projektów rozwiązuje problem tak specyficzny, że żadna firma nigdy by się nim nie zajęła. Nie ma w tym biznesu. Ale dla jednej osoby, dla jednej rodziny – to może być dokładnie to, czego potrzebowali. Aplikacje domowej roboty, szyte na miarę.

Ołówek, nie różdżka

Ale żeby nie było nieporozumień: vibe coding nie oznacza, że AI robi wszystko za Ciebie.

Scott H. Young, autor „Zostań ultrasamoukiem”, ujął to trafnie: „Świat, w którym każdy może kodować, to prawdopodobnie świat z mnóstwem szmiry – ale też ze znacznie wyższą poprzeczką dla tego, co naprawdę wartościowe. Vibe coding przerzuca ciężar na ludzi: żeby mieli lepszą wizję, lepsze rozumienie koncepcyjne i lepszy gust”.

Bo AI to tylko narzędzie. Jak ołówek. Tym samym ołówkiem jeden narysuje arcydzieło, a drugi – ludzika z kilku kresek. Ołówek nie wie, co chcesz narysować. Nie wie, czy ma powstać portret czy karykatura, szkic techniczny czy komiks. Ty to wiesz.

Podobnie było z moją grą. Czy wpisałam „zrób mi fajną grę RPG” i dostałam gotowy produkt? Absolutnie nie. Zaczęłam od projektu: rozłożyłam wizję na części składowe, określiłam, co dokładnie chcę osiągnąć i jak poszczególne elementy mają do siebie pasować. „Hexbreak” to powieść wizualna – gatunek gry opowiedzianej przez dialogi, obrazy i wybory gracza, z fabułą o księżniczce i czarodzieju, zakazanej magii i politycznych intrygach. Żeby ją zbudować, potrzebowałam autorskiego silnika, systemu rozgałęzień fabularnych, grafik, muzyki, dubbingu i porządnej dawki pracy nad interfejsem. Bo vibe coding to tylko jedna warstwa. Obok niego istnieje dziś cały ekosystem narzędzi, które robią z opisów słownych rzeczy jeszcze niedawno zarezerwowane dla profesjonalistów: ElevenLabs zamieniło tekst w realistyczny dubbing, Udio skomponowało muzykę na podstawie kilku zdań, a modele Stable Diffusion wygenerowały grafikę.

Nie będę udawać, że szło bez zgrzytów. Silnik pisałam we Flutterze – języku, którego wcześniej nie znałam. Jeśli AI wygenerowało jakąś bzdurę, nie byłam w stanie ocenić poprawności kodu i musiałam polegać na testach funkcjonalnych. A grafika? Postać musi wyglądać tak samo na setkach ujęć, tymczasem modele obrazowe za każdym razem rysują ją odrobinę inaczej, więc potrzeba było wielu manualnych poprawek.

Narzędzia AI tworzyły elementy składowe, ale to ja pisałam intencję – fabułę, logikę, architekturę.

Programiści nie znikną – ale zmienia się to, czym właściwie jest programowanie. Przez dekady zdolność napisania kodu była warunkiem wejścia do świata technologii. Jeśli miałeś pomysł, musiałeś najpierw nauczyć się języka maszyny albo znaleźć kogoś, kto go znał. Kod był wąskim gardłem między wizją a realizacją.

Dziś po raz pierwszy ta przeszkoda zaczyna zniknąć. AI przejmuje dużą część odpowiedzi na pytanie „jak”, zostawiając ludziom pytania trudniejsze: „co” i „po co”. Zamiast pisać każdą linijkę kodu, twórca coraz częściej staje się kimś w rodzaju dyrygenta – określa kierunek, podejmuje decyzje, pilnuje spójności. CEO firmy Anthropic ujął to w ten sposób: „Za rok AI będzie pisało większość kodu, zostawiając ludziom marzenie na większą skalę”. Brzmi górnolotnie, ale w praktyce chodzi o to: jeśli nie musisz spędzać godzin na składni i debugowaniu, możesz więcej czasu poświęcić na decyzje, które naprawdę kształtują projekt.

To nie znaczy, że umiejętności techniczne przestają mieć znaczenie. Wręcz przeciwnie. Za to przestają być jedyną drogą do tworzenia. Coraz częściej o jakości projektu decyduje nie to, kto najszybciej napisze kod, lecz kto lepiej rozumie problem, potrafi zaprojektować system i ma wystarczająco dobre wyczucie, by odróżnić rozwiązanie działające od rozwiązania właściwego.

AI potrafi wygenerować kod. Nadal nie potrafi zdecydować, co warto zbudować.

Demokratyzacja tworzenia

Przez dziesięciolecia istniała przepaść między wizją a realizacją. Żeby ją pokonać, trzeba było lat nauki, zespołu specjalistów albo rezygnacji z ambicji na rzecz czegoś prostszego. Teraz ta przepaść się kurczy. Nie znika – nadal trzeba wiedzieć, co się chce zbudować, i potrafić spojrzeć na efekt krytycznym okiem. Ale próg wejścia gwałtownie spadł.

Jensen Huang, CEO Nvidii, nazywa AI „wielkim wyrównywaczem”. Kiedyś – mówił – trzeba było znać C++ albo Pythona, projektować skomplikowane systemy, uczyć się języka maszyn. Teraz pojawił się nowy język programowania: język ludzki. „Niewielu ludzi zna C++, jeszcze mniej zna Pythona – ale każdy zna język ludzki”.

Po raz pierwszy w historii informatyki to nie człowiek uczy się języka maszyny – to maszyna nauczyła się języka człowieka.