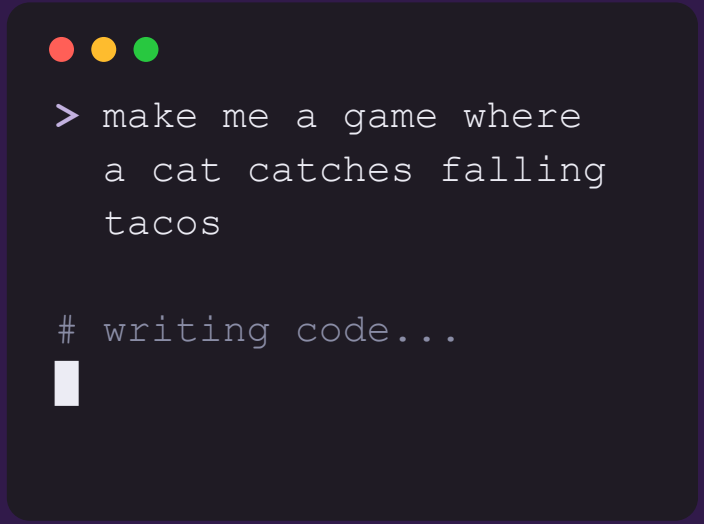
AI Vibe Coding

Building apps and games by describing what you want



ACP Middle School Field Day · Fall 2026

Quick poll: raise your hand if you have ever…



Played a video game



Used an app on a phone



Made a game or app yourself

Every app you use is made of code, written line by line

```
# simplified game code
cat_x = 200
score = 0
while playing:
  if key_pressed("left"):
    cat_x = cat_x - 5
  if taco.touches(cat):
    score = score + 1
  draw(cat, cat_x, 400)
```

Code is exact. One missing symbol and nothing runs.

It takes practice. People spend months or years learning to write it.

Real apps are huge. Big ones have millions of lines.

Python turns your code into bytecode: tiny steps the computer can follow

```python
# loop.py
total = 0
for i in range(3):
    total = total + i
```



```
    LOAD_CONST    0
    STORE_NAME    total
    LOAD_NAME     range
    PUSH_NULL
    LOAD_CONST    3
    CALL          1
    GET_ITER
L1: FOR_ITER      to L2
    STORE_NAME    i
    LOAD_NAME     total
    LOAD_NAME     i
    BINARY_OP     +       # add!
    STORE_NAME    total
    JUMP_BACKWARD to L1   # again
L2: END_FOR
    POP_TOP
```

3 lines become 16 steps

Each step is tiny: load a value, add two numbers, store the result, jump back to the top.

Under it all, the processor only runs numbers, stored as 1s and 0s

the loop from a C version of this program, compiled for an Intel/AMD (x86-64) chip

```
# assembly                        # machine code     # binary
mov eax,[rbp-0x4]                 8b 45 fc         10001011 01000101 11111100
add [rbp-0x8],eax                 01 45 f8         00000001 01000101 11111000
add DWORD PTR [rbp-0x4],0x1       83 45 fc 01      10000011 01000101 11111100 00000001
cmp DWORD PTR [rbp-0x4],0x2       83 7d fc 02      10000011 01111101 11111100 00000010
jle 0x18                          7e f0            01111110 11110000
```

Assembly

Instructions people can read

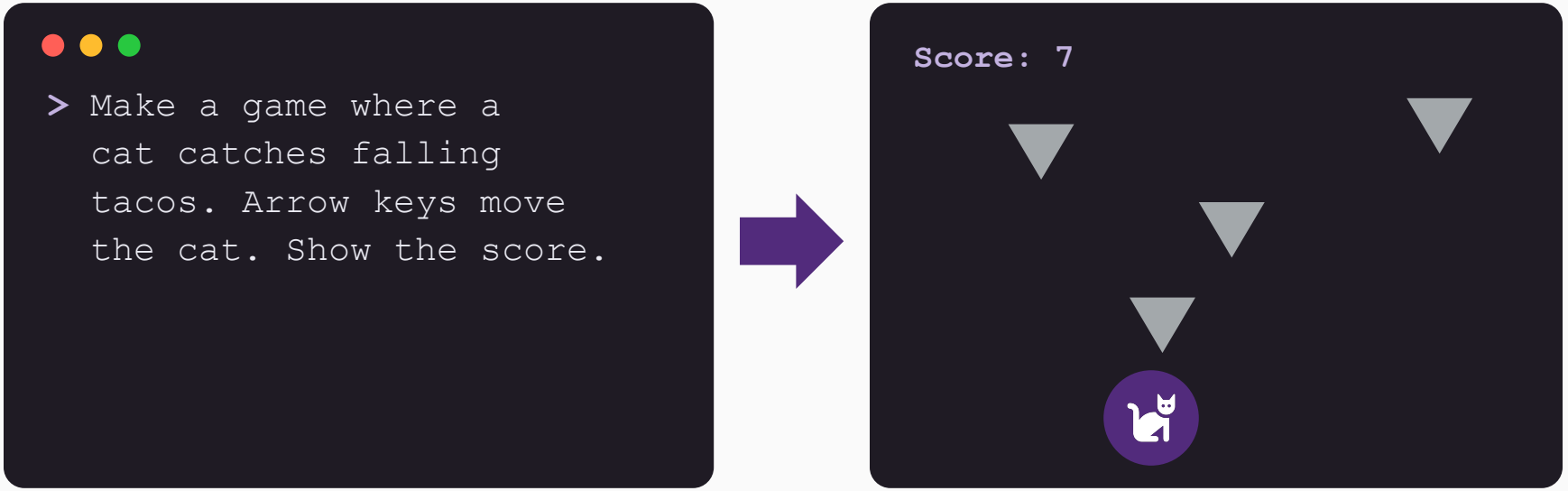
Machine code

Those instructions as numbers

Binary

The numbers as 1s and 0s

Now you can describe what you want in plain English



The AI writes all of the code. You test the game and ask for changes.

In 2025, this got a name

vibe coding

Coined by AI researcher Andrej Karpathy in February 2025[1]

Named Collins Dictionary's Word of the Year for 2025[2]

1. Andrej Karpathy, post on X, Feb. 2, 2025. 2. Collins Dictionary, Word of the Year announcement, Nov. 6, 2025.

PART 1

How it works

Describe it. Build it. Try it. Fix it.

Vibe coding is a loop you repeat until it's right



Describe

Say what you want in plain words



Build

The AI writes the code



Try

Run it and play with it



Fix

Say what's wrong or what to add

Repeat until it does what you imagined

The AI is a super-fast typist. You are the boss.



The AI does

- Writes the code
- Fixes error messages
- Explains what code does



You do

- Decide what to build
- Test it and notice problems
- Describe what "good" looks like
- Decide when it's done

PART 2

Let's build something

Live, right now. Things might break. That's part of it.

You design the game: pick one from each column

1 Hero	2 Catch or dodge	3 Twist
A cat	Falling tacos	Speeds up over time
A robot	Asteroids	Gravity flips
A dinosaur	Flying homework	A boss appears

Step 1: one sentence is enough to start

```
> Make a game I can play in a web browser. A dinosaur
  at the bottom of the screen dodges falling asteroids.
  Arrow keys move it. Show the score, and speed up
  over time.
```

What kind of thing

a browser game

What's in it

a dinosaur, asteroids

How you play

arrow keys, score, speed

Step 2: play it, then ask for one change at a time

```
> Make the asteroids glow orange
```

```
> Add a high score that stays after I refresh
```

```
> When I lose, show GAME OVER and a restart
```

```
> Add a sound when I get hit
```

```
> Make it look like an old arcade machine
```



One change per message

If something breaks, you know exactly which change caused it.

When it breaks, describe what you see and what you expected

```
> When I hold the left
  arrow, the dinosaur
  goes off the edge and
  disappears. It should
  stop at the wall.
```

What I did

held the left arrow

What happened

it disappeared

What should happen

stop at the wall

PART 3

Be the AI

A game where you are the computer

How to play: one person describes, one person draws



The programmer

Sees the secret picture. Describes it using only words. No pointing or hand motions.



The AI

Stands at the board, back to the screen. Draws exactly what they hear, nothing more.

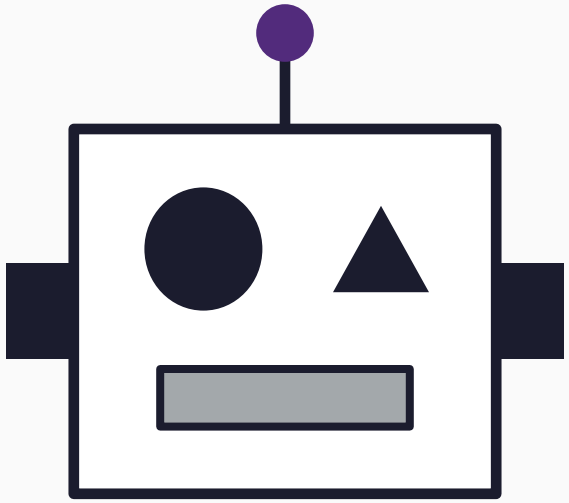


Everyone else

Stays silent and watches what goes wrong.

Round 2: the programmer can look at the drawing and give fixes, like you would with real AI.

Secret picture: AI, don't look!



Programmer tips

- Say shapes, not things: "a big square," not "a head"
- Say where: "in the middle," "at the top"
- Say how big compared to other shapes

Being exact is the real skill, with or without AI



Be specific

Shapes, sizes, positions and colors beat vague words like "a robot."



Check the result

Compare what you got to what you pictured.



Fix one thing at a time

Small, clear changes are easy to check.

PART 4

What could you build?

Ideas, rules, and how to start

Things you could build this year



A game for your friends



A website for your club or team



A quiz to study for a test



A tracker for your sports stats



A tool for a family member's job



An animated birthday card

AI makes mistakes, so you stay in charge



Test everything

AI can sound sure and still be wrong.



Keep private info private

Never type passwords, your address or personal details.



Follow your school's rules

Ask your teacher before using AI on schoolwork.



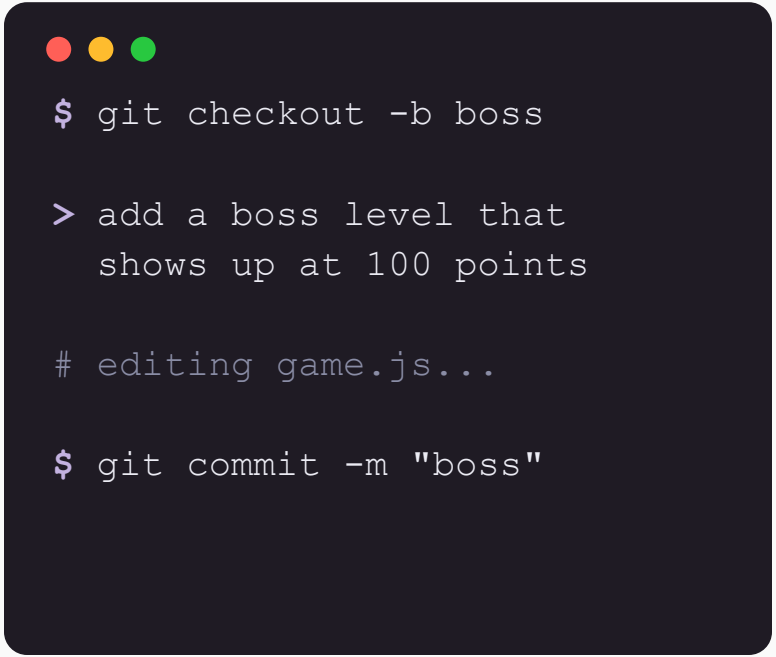
Check the age rules

Most AI tools need you to be 13+ with a parent's OK. Some are 18+.

Source: OpenAI Help Center (ChatGPT: 13+, parent consent under 18); Google Gemini age requirements (13+); Anthropic Consumer Terms (Claude.ai: 18+).

Already code? AI makes your skills worth more

- Read the code the AI writes, and ask it to explain lines you don't know
- Learn git so you can undo any change
- Try AI coding agents that work in a terminal and edit whole projects
- Build something bigger than you could ever type alone



Start tonight: three ways to try it



Learn how programs think

Scratch (**scratch.mit.edu**) is free and made for your age. No AI needed.



Ask AI for a one-file game

With a parent, try:

- **Claude Code**
- **OpenAI Codex**
- **Google AI Studio**



Make it 10 changes better

One change at a time. Save each version you like.

THANK YOU

What will you build?

Questions?

Appendix: demo prompts in order

```
> Make a single-file HTML game I can open in a browser. A [hero] at the
  bottom dodges falling [things]. Arrow keys move it. Show the score.
> Make it speed up every 10 seconds.
> Add a high score saved in the browser.
> When I lose, show GAME OVER and a restart button.
> Make it look like an old arcade machine, with sound effects.
> Add [twist the class picked].
```