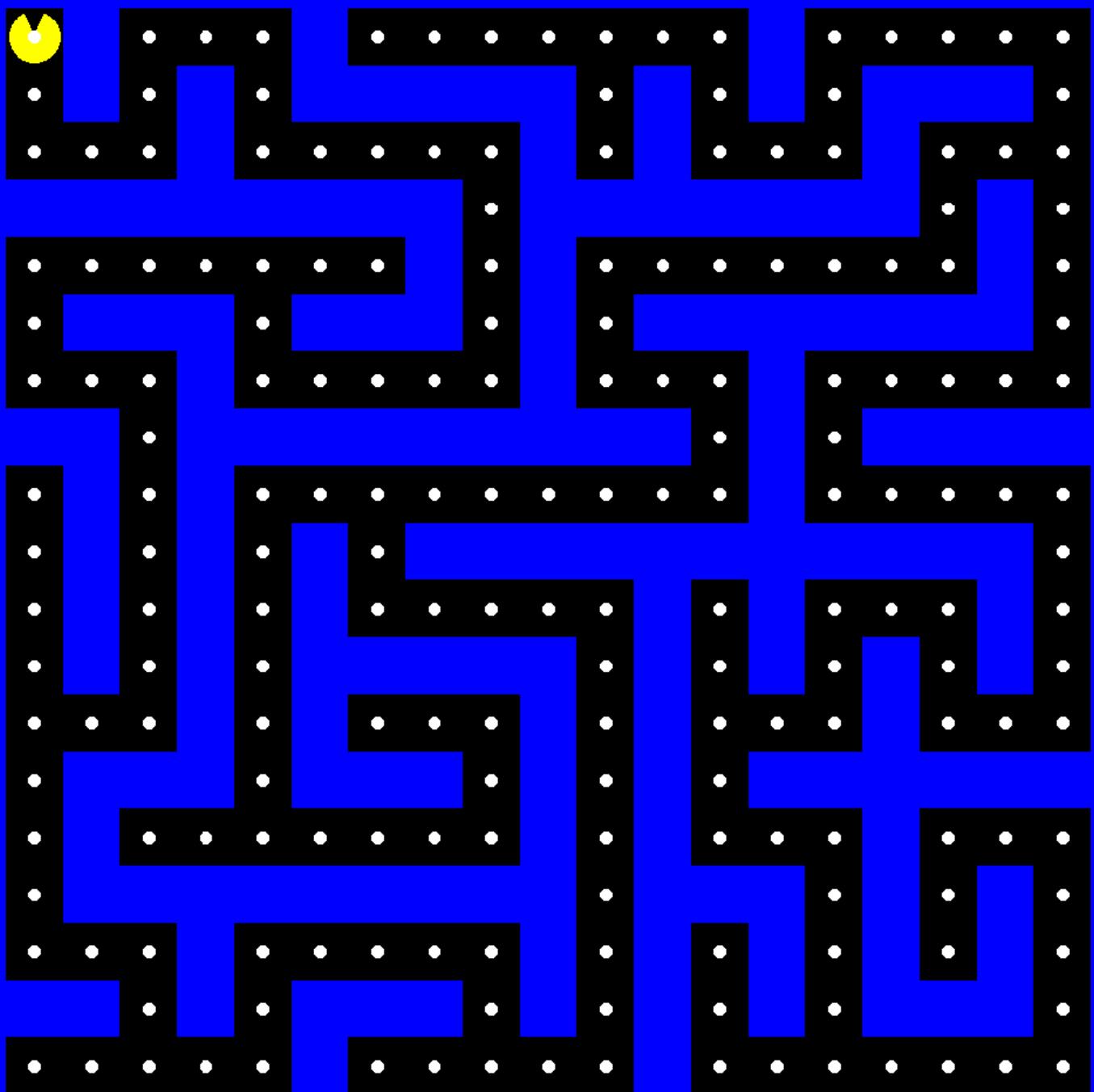


ChatGPT + Code Interpreter = Magic

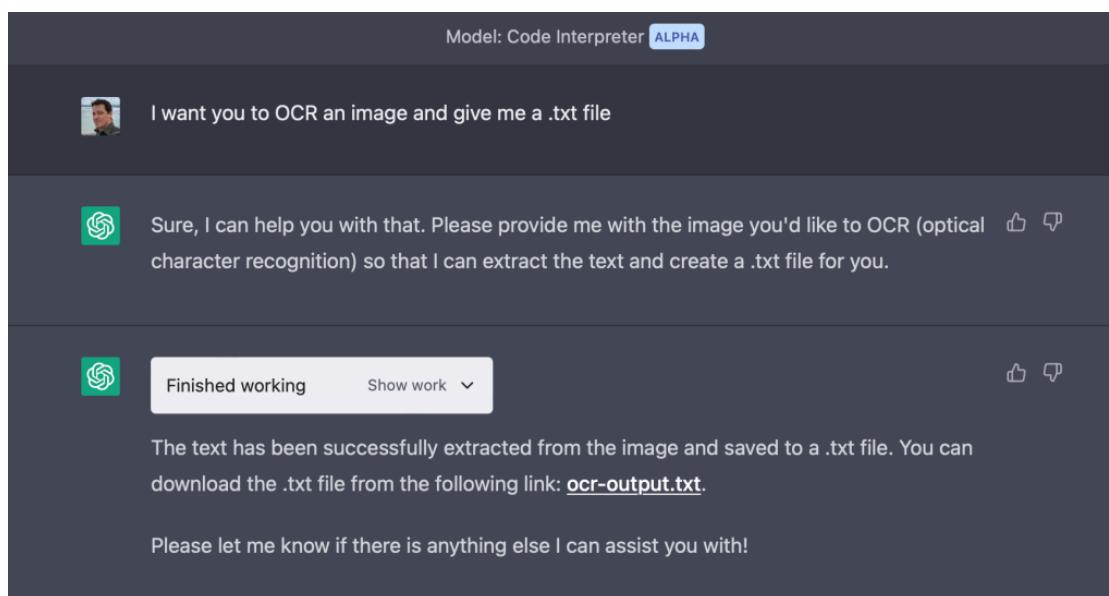
• Andrew Mayne ⏰ March 23, 2023March 23, 2023 ⏷ 5 Minutes



tl;dr: OpenAI is testing the ability to run code and use third-party plugins in ChatGPT.

OpenAI has announced that we're developing plugins for ChatGPT that will extend its capabilities. [[Link](#)] (<https://openai.com/blog/chatgpt-plugins>) Plugins range from third-party tools like *WolframAlpha* and *OpenTable*, to our browsing plugin and *Code Interpreter* that can generate code, run code, upload and download files ranging from csv data to images and evaluate the output all within the ChatGPT interface.

Currently the *Code Interpreter* runs Python with a small selection of libraries. I've been playing with *Code Interpreter* and it's been a lot of fun to see what it can do even with basic libraries.



Beside generating code, *Code Interpreter* (CI) can analyze the output and use it in another function. This means that you can string together different sections of code, taking the output of one and feeding it to another. The Pac-Man gif above was made by having CI use an algorithm to generate a maze, convert the maze into blocks, use an algorithm to find the exit, make it look like Pac-Man and then generate a gif.

Previously when I used ChatGPT to create code this involved taking the output and putting it into another environment to test it. Now you can do a lot of development inside ChatGPT without leaving the UI. When you add that to its ability to analyze data, such as spreadsheets, and then doing functions like graphing, you have a very powerful tool for both coding and research.

Here are some examples of random experiments I ran playing around with ChatGPT's Code Interpreter plugin.

Sound

When I realized that ChatGPT understands a lot of mathematical data and CI can generate downloadable files, I asked ChatGPT to generate a Shepard tone (a sound illusion that seems like it's constantly getting higher in pitch.) ChatGPT showed the tone in a graph and then was able to output it as a wav using CI:

Model: Code Interpreter ALPHA

 Create a Shepard tone and save it as a wav

 A Shepard tone is an auditory illusion that creates the impression of a tone that continuously ascends or descends in pitch, but never seems to get any higher or lower. It consists of several sine waves, each an octave apart, that are overlapped and looped. The amplitude of each sine wave changes over time in a way that creates the illusion.

To create a Shepard tone, we will:

1. Define a range of frequencies, each an octave apart.
2. Generate sine waves for each frequency.
3. Modulate the amplitude of each sine wave to create the illusion of a continuous ascending or descending pitch.
4. Combine the sine waves to create the Shepard tone.
5. Save the tone as a .wav file.

Let's create a Shepard tone that gives the illusion of continuously ascending pitch. We'll start by defining some parameters and generating the sine waves.

Finished working Show work ▾

The Shepard tone has been successfully generated and saved as a WAV file. You can download the ascending Shepard tone from the following link:
[ascending_shepard_tone.wav](#).

Shepard tone

0:00 / 0:10

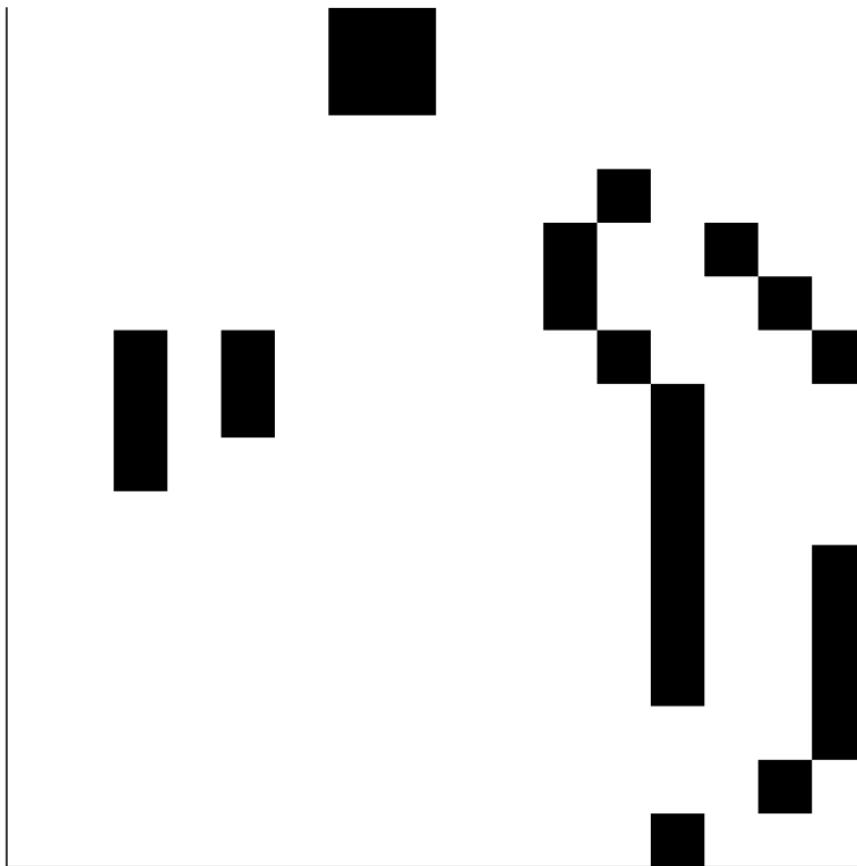
Pushing the sound capability a little further, I asked ChatGPT to generate a session of Conway's Game of Life and then turn it into music. When the first pass sounded a little too random, I asked it to make it more pleasing. ChatGPT applied a pentatonic scale to it and this was the result:

Game of Life in Song

0:00 / 0:14

Imagining an entire album of musical math (Fibonacci sequences, Mandelbrot sets, Cellular Automata and more!) I asked ChatGPT to generate an album cover for the Game of Life song. Since it's not connected to DALL-E, I asked it to generate the image from a frame of the simulation.

Game of Life frame (as album art)



QR Codes

Looking at the image made me realize that ChatGPT could probably generate a functioning QR code image. I gave it a simple one and here is the result:



Create a QR code for OpenAI.com



Finished working

Show work ▾



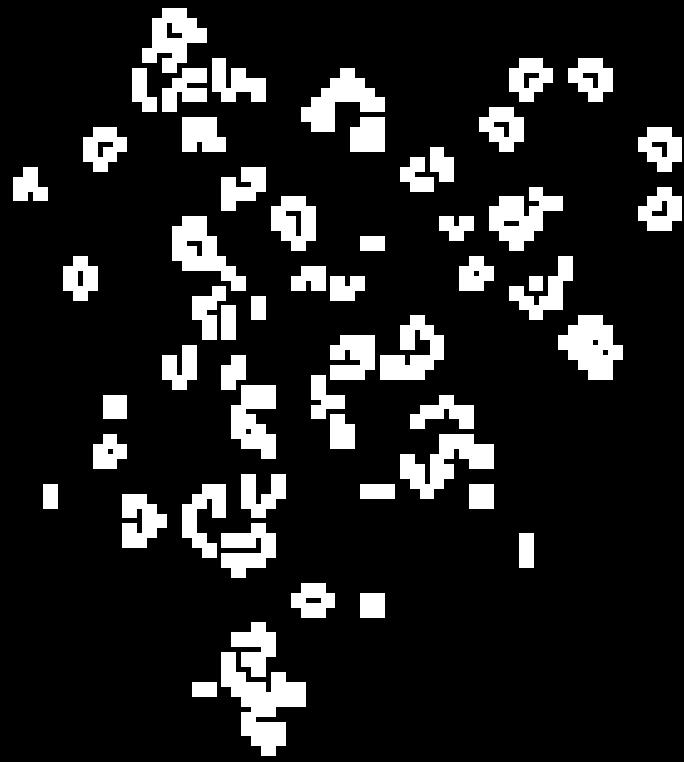
The QR code for the website "<https://www.openai.com/>" has been generated and displayed. If you'd like to save the QR code as an image file, please let me know the desired file format (e.g., PNG or JPEG) and I'll be happy to provide the file for you.

QR code



Looking at the Game of Life and QR code next to each other had me wondering...could I get ChatGPT to create a Game of Life simulation that ended in a QR code? (I told it to cheat by working backwards from the QR code...) Yes, it can:

Game of Life QR code



In the blog post [link] on ChatGPT plugins we show some cool use of Python libraries to do image manipulation. This made me curious to see what else you could do with basic libraries.

Using OpenCV and a simple face finding algorithm (Haar Cascade Classifier) , CI was able to create a function to find my face in this image:

Face finding with OpenCV



Since it can open and read a file, that also means that it could in theory use that data to generate something new. So I asked ChatGPT to take an image and convert it to an ASCII image:

Uploaded image

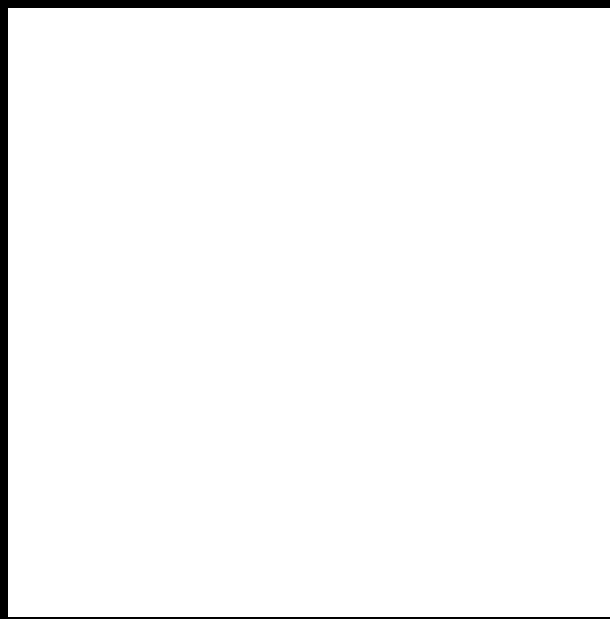


Converted to ASCII

Generative Animation

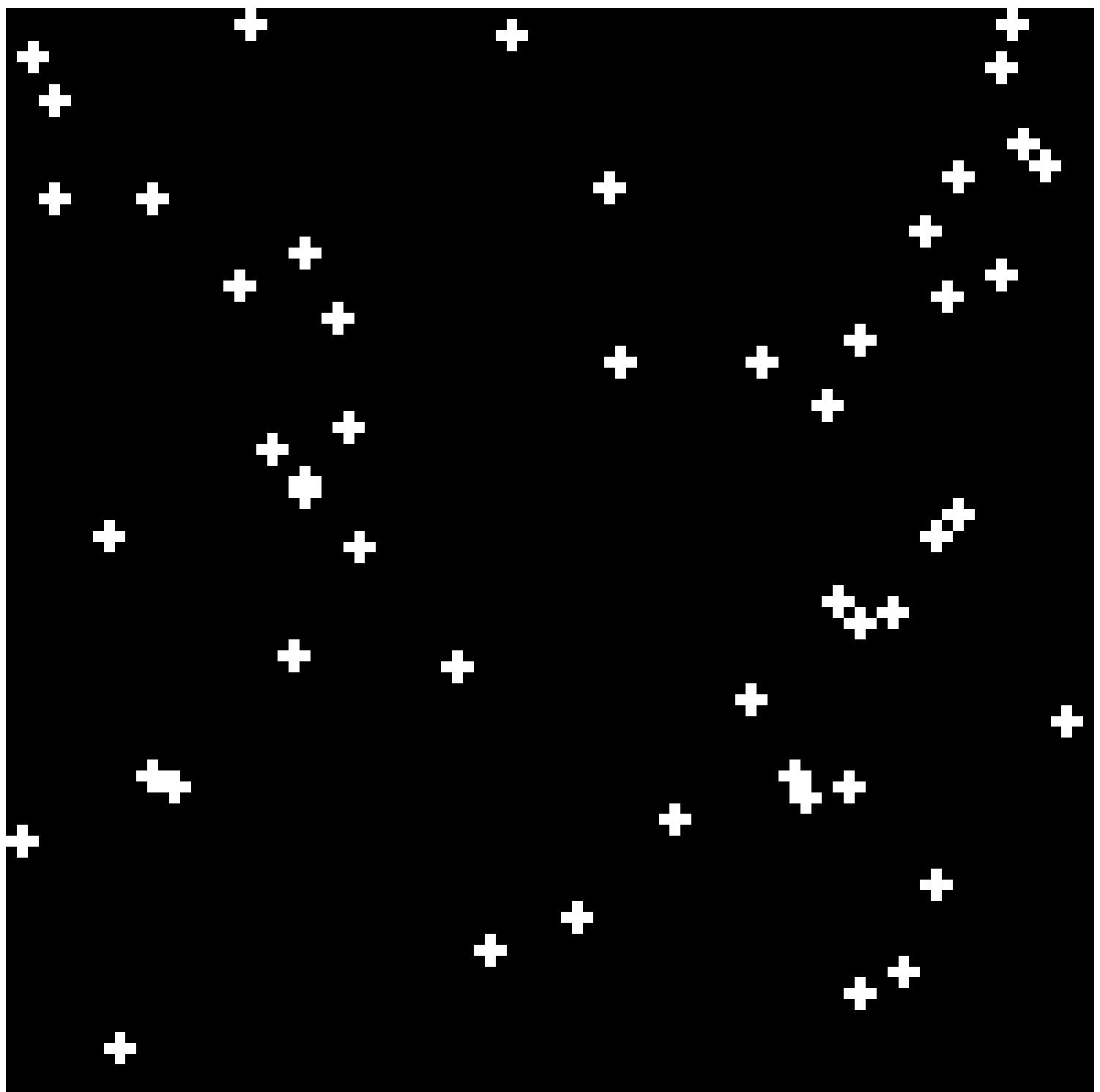
The fact that CI could generate data and output it into different forms including animated gifs had me wondering how far you could push this. First I asked it to make a bouncing block:

Bouncing block



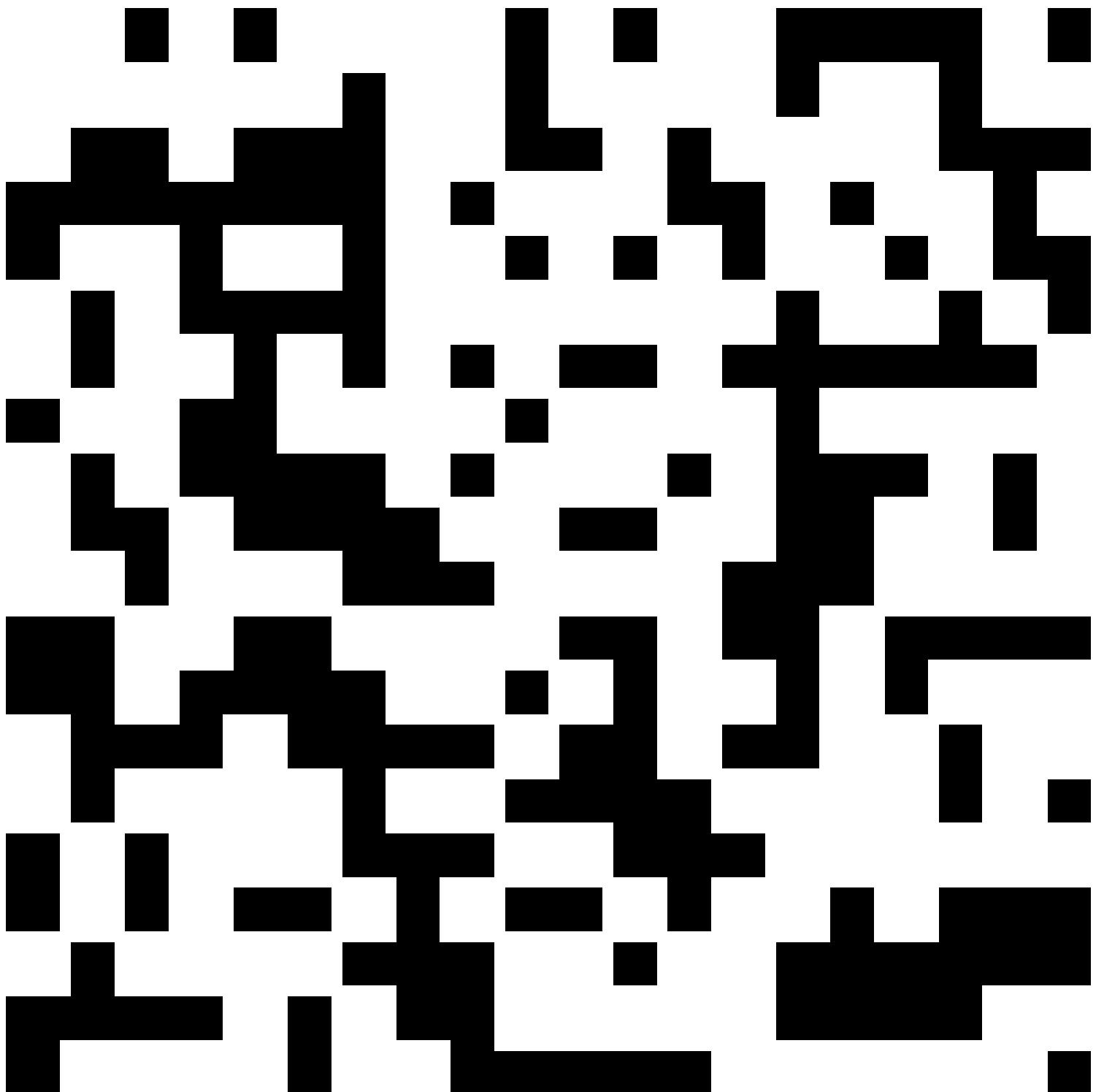
Then a snowstorm:

Snowstorm



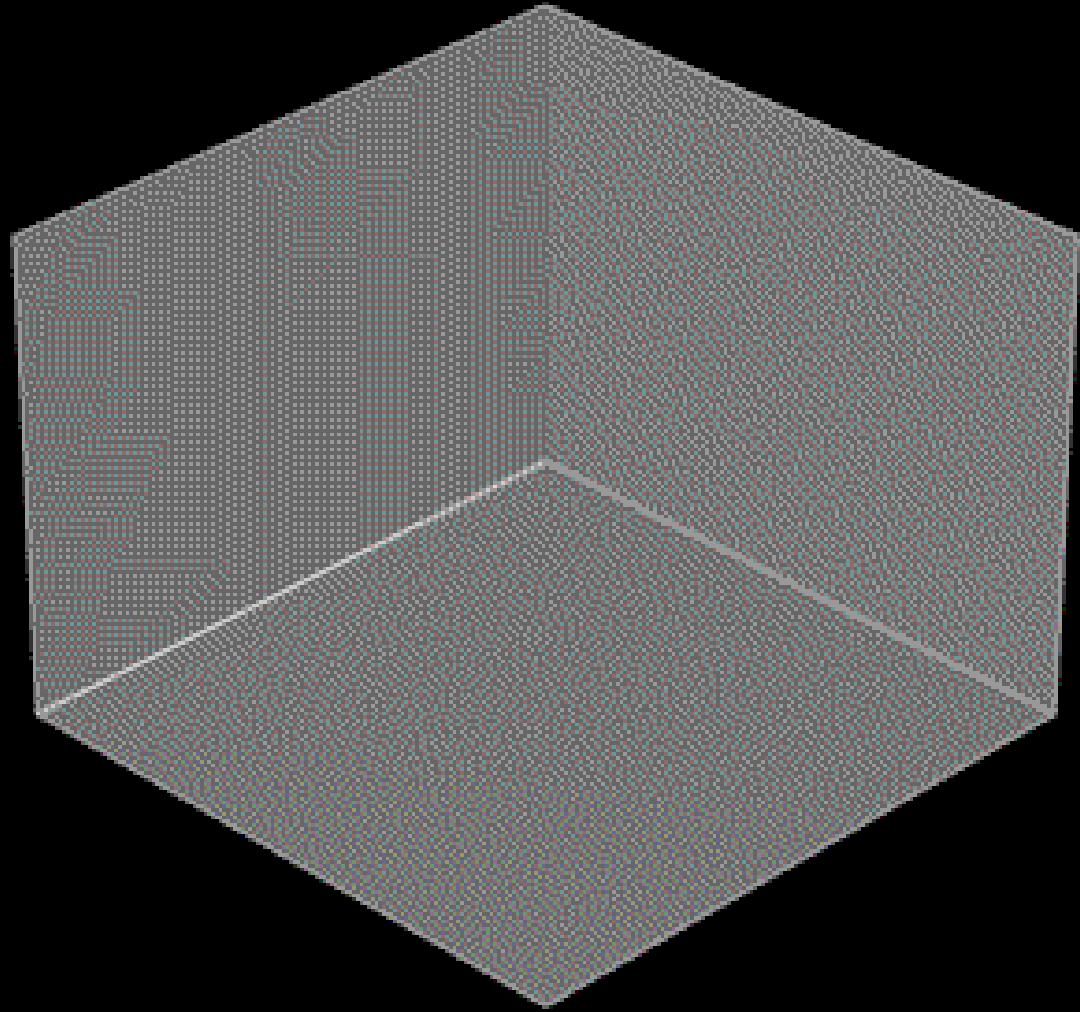
Next a session of Game of Life (this is what inspired the QR code experiment):

[Game of life]



I tried to get it to do this with a vectorized cube. This is as close as I got before trying something else:

[Rotating cube]



OCR

It turns out that Optical Character Recognition is one of the libraries in CI. I decided to test this out with a screen grab from a website. Sure enough, it works. The ability to combine OCR with a really capable language model introduces some interesting possibilities for restoring older documents.

At one point, a buoy bobbing atop the ocean near Oregon detected frightening jumps in temperature of up to seven degrees Celsius in less than an hour. The ocean was sweltering.

But scientists, with their attention fixed on temperature data streaming in from ocean surfaces, had little idea what was transpiring in the depths below.

Now, new modeling led by researchers at the US National Oceanic and Atmospheric Administration (NOAA) shows that marine heatwaves can unfold deep underwater, too – sometimes in tandem with heatwaves that ripple across the ocean surface or else when there is no detectable warming signal above.

The new analysis, of continental shelf waters surrounding North America, also finds these so-called bottom marine heatwaves can be more intense and last longer than hot spells at the ocean surface, though it varies from coast to coast.

"Researchers have been investigating marine heat waves at the sea surface for over a decade now," says lead author Dillon Amaya, a climate scientist with NOAA's Physical Science Laboratory.

Calendar invites

Although CI doesn't have access to the ics library, it can generate a calendar invite in plain text and then have ChatGPT save it as an .ics file.

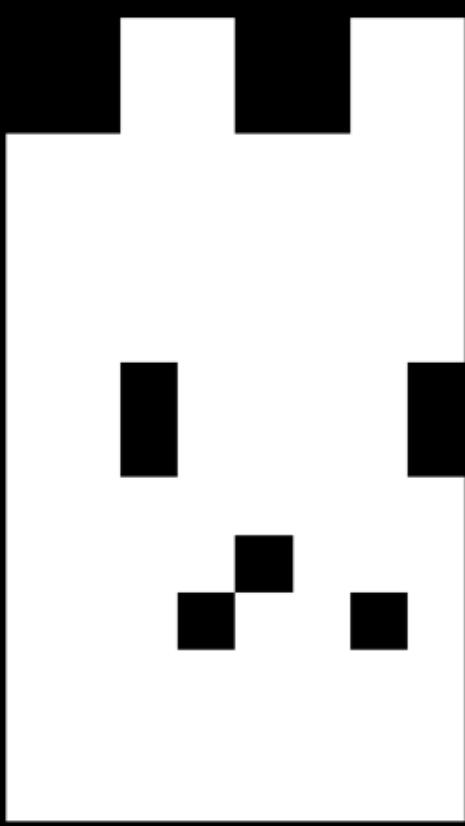


Celebrate May
the 4th

Drawing

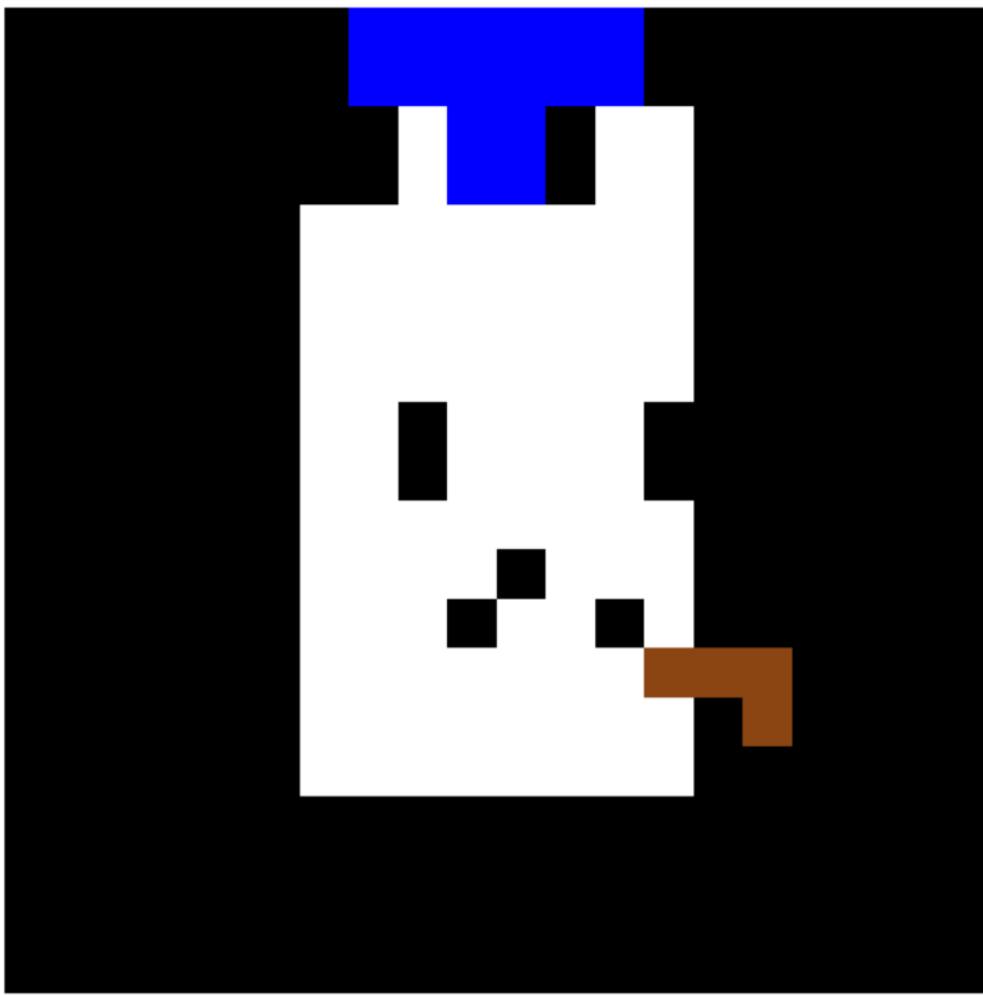
While there's been some interesting examples of GPT-4 generating SVGs, I was curious to see what ChatGPT and IC could do with simple block images. I asked it to draw a cat and it created something cat-like. Just to make sure it wasn't doing that from memory, I asked for a blue top hat and pipe. While not exactly the top hat I had in mind – still a hat and blue.

A plausible cat



A plausible cat with a hat and a pipe

Cat with Blue Top Hat and Pipe



Here is the updated image of the cat, now wearing a blue top hat and smoking a pipe:

A cat in the hat with a pipe



Can you give it a hat like "The Cat in the Hat"?

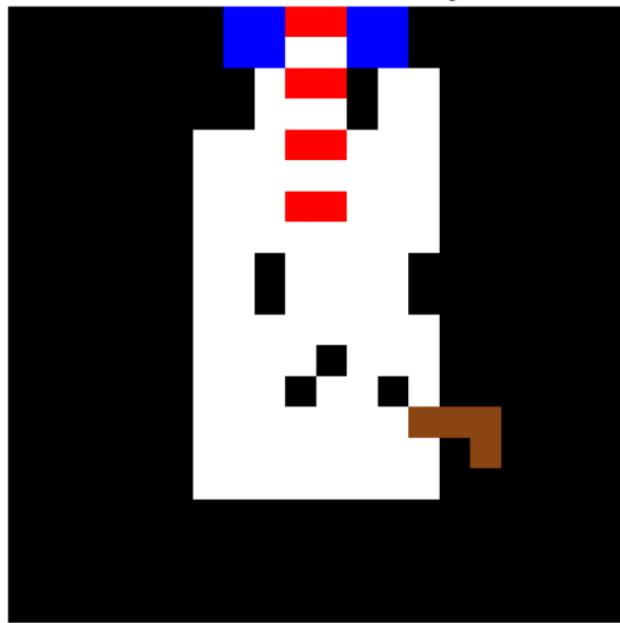


Finished working

Show work ▾



Cat with 'The Cat in the Hat' Style Hat and Pipe



Here is the updated image of the cat, now wearing a hat in the style of "The Cat in the Hat" and smoking a pipe:

Creating machine learning models (sort of)

While ChatGPT can't load any machine learning libraries into CI (presently), we can do some basic statistical text generation using n-grams. In this example I gave it one of my books, asked it to create a prediction algorithm and then predict the next word in a text sequence. The results won't have GPT-4 worried anytime soon, but were still interesting to see.

Next word prediction



Can we build our own machine learning model from scratch here?



Yes, we can build a simple machine learning model from scratch using Python. What kind of problem do you want to solve with the model? ⏪ ⏴



Next word prediction



Next word prediction is a natural language processing task that can be tackled with a language model. One simple approach to building a language model is to use an n-gram model.

An n-gram model is a statistical language model that predicts the next word based on the previous n-1 words. For example, if we choose n=3 (i.e., a trigram model), the model would predict the next word based on the previous two words.

Here's some code to get you started with building a trigram model from scratch:

```
python
Copy code

from collections import defaultdict, Counter

def train_trigram_model(corpus):
    """
    Train a trigram model on the given corpus.

    Args:
        corpus (str): A string containing the text corpus for training.

    Returns:
        A dictionary where keys are tuples of two words (w1, w2) and values
    """
    model = defaultdict(Counter)

    # Tokenize the corpus into words
    words = corpus.split()

    # Iterate over the words
    for w1, w2, w3 in zip(words[:-2], words[1:-1], words[2:]):
        model[(w1, w2)][w3] += 1

    return model
```





Sure, go ahead! You can use the `predict_next_word` function to predict the next word based on a sequence of words. Just provide a list of words as input to the function.



They went to the



Finished working

Hide work ^



python

Copy code

```
input_words = "They went to the".split()  
predicted_word = predict_next_word(ngram_counts, input_words)  
predicted_word
```

RESULT

'shoulders'

Here's the input sequence of words:

"They went to the"

And the predicted next word using the n-gram model is:

"shoulders"

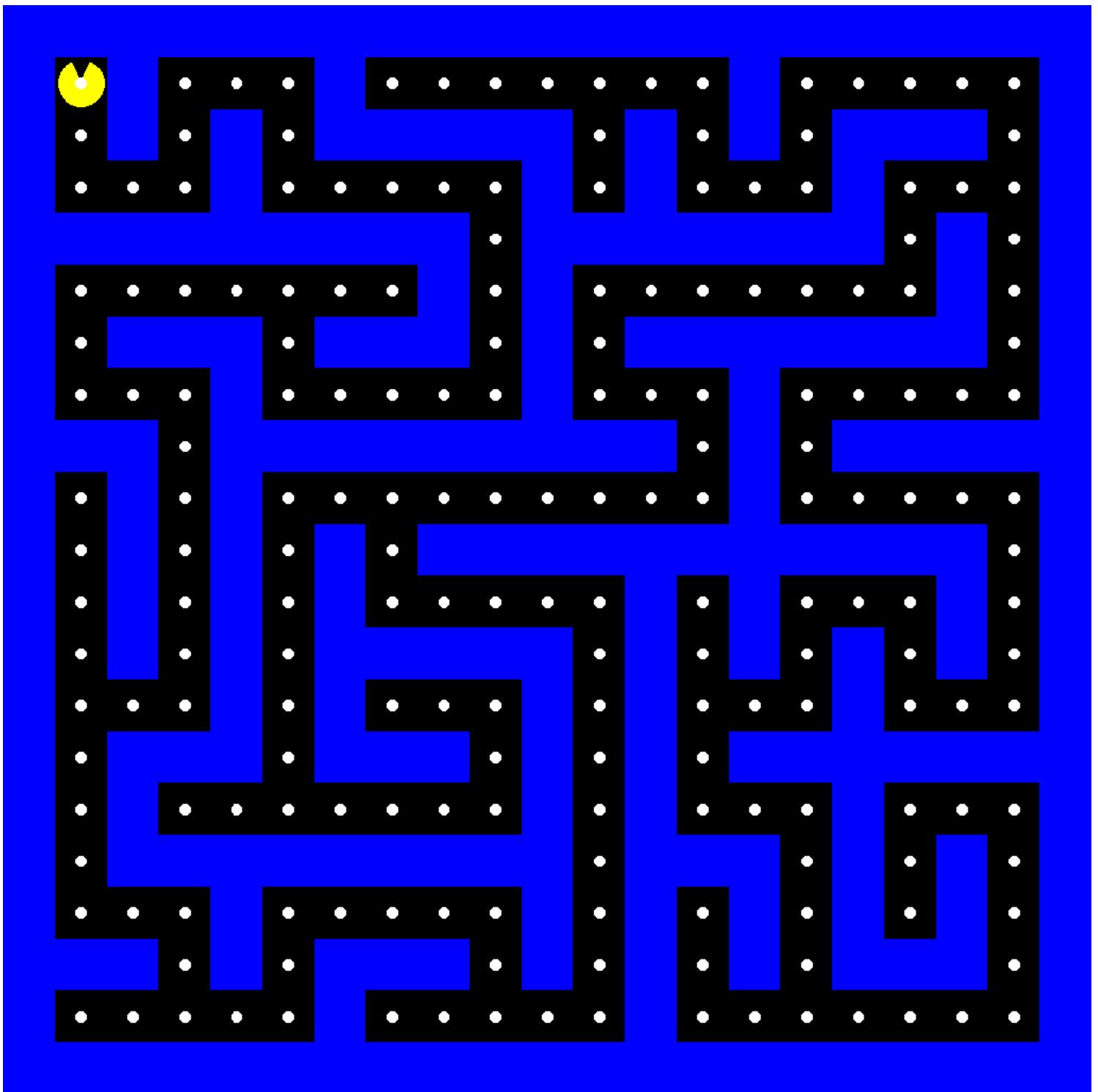
So, the complete sequence becomes:

"They went to the shoulders"

Simulation

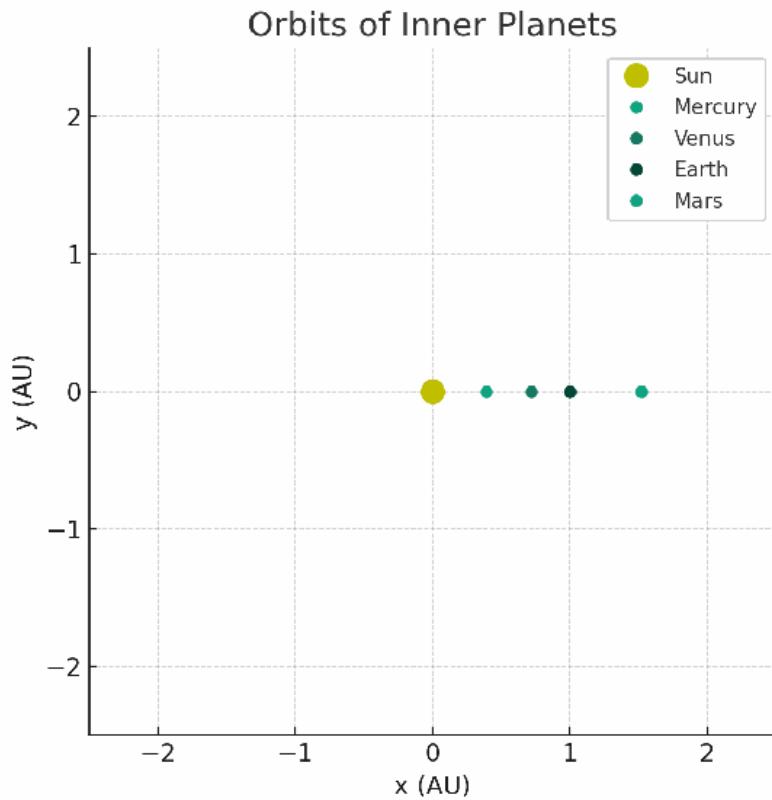
I decided to revisit CI's ability to generate animated gifs and algorithms and had it make a maze, then have a dot find its way out of the maze...then make the whole thing look like Pac-Man:

Pac-Man simulator



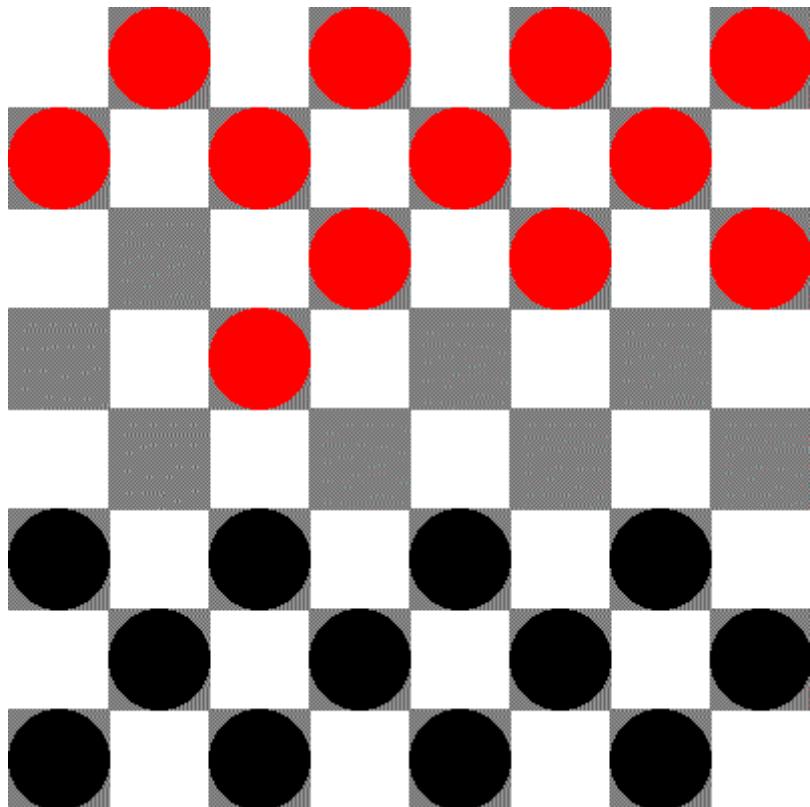
Using the Matplotlib library I was able to get ChatGPT to make a simulation of the orbits of the inner planets and save it as a gif.

Orbits of planets



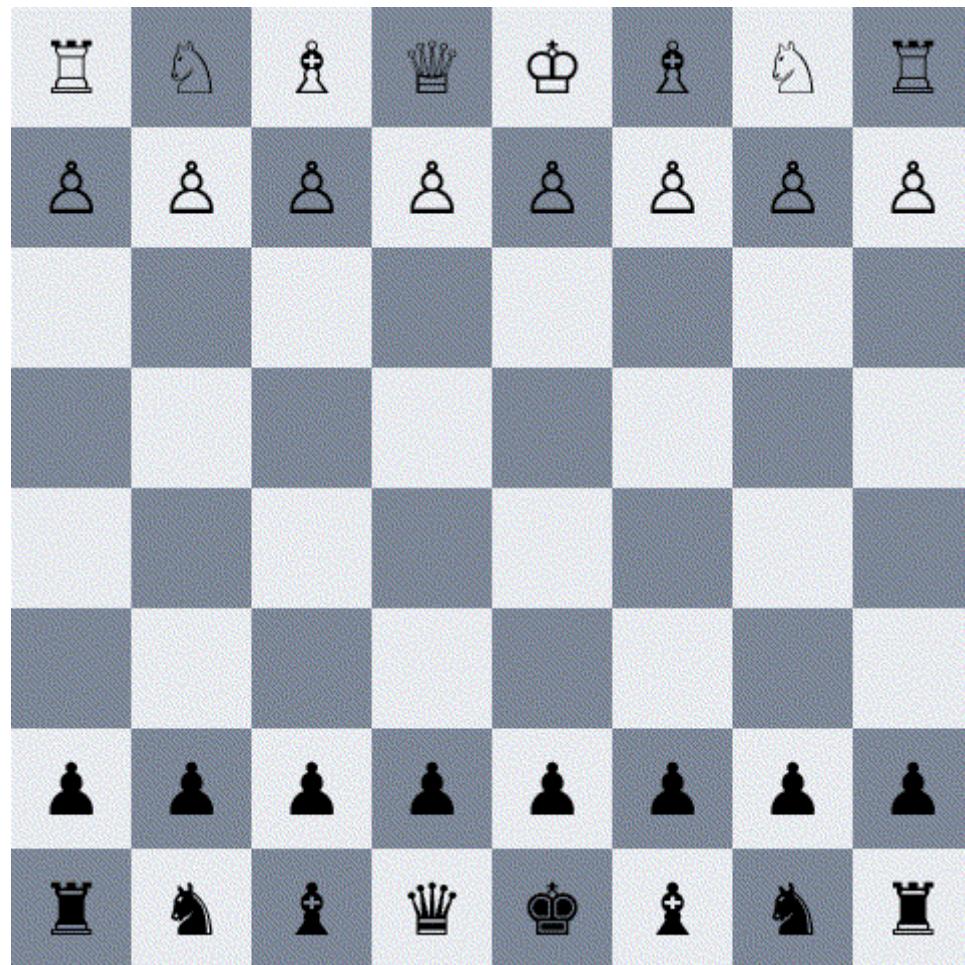
I then wondered if I could get it to simulate a game of checkers. I managed to move the pieces, but was too polite to take any of the other players' checkers:

Checkers



What about chess? First I had to get ChatGPT to create its own chess library (a very simple chess library of reduced and not quite legal moves...) and use unicode for the pieces, after that, it was able to simulate the start of a game and save it as a .gif:

Chess



Here's the conversation I had with ChatGPT to make the chess simulation. As you can see, I just gave a few instructions and ChatGPT and CI did the rest:

Wrapping up

These are just some of my explorations with ChatGPT and the Code Interpreter plugin. As more people get access I'm sure we're going to see some really mind-blowing applications.

Tagged:

CHatGPT,
Code,
OpenAI

Published by Andrew Mayne



WSJ best selling novelist, Edgar & Thriller Award finalist, star of Shark Week, A&E Don't Trust Andrew Mayne, creative applications and Science Communicator at OpenAI. [View all posts by Andrew Mayne](#)

[Website Powered by WordPress.com.](#)

