

*AI Recitation Lecture 2/12-2024*

# How to Make (Almost) Anything (Almost) without Making Anything

*Olivia Seow, Harvard University*

*Amira Abdel-Rahman, MIT Center for Bits and Atoms*

*Valdemar Danry, MIT Media Lab*



**Poonam Soni** @CodeByPoonam · Sep 17

AI Jobs can pay a salary of \$375,000 per year.

If you aren't



**Insider Business** @BusinessInsider · Sep 14

Employees are increasingly getting 'AI Anxious' — they're worried about **falling behind** on AI and even pretending to be 'in the know' to colleagues

**Companies and int...  
expertise will be le...**

BY OLAFIMIHAN OSHIN - 05/28/23 7:!!



**JD's Coke Habit** @JDsBadHabit · Sep 11

If you aren't using AI, you're **falling behind**. Here's my top 5 ways to utilize this tool now:



**Zain Kahn** @heykahn

If you're not using AI, you're falling behind.

Member-only story

**Using AI won't make y...  
cheater — but not usi...  
leave you behind**

And why people using AI will be the one replacing you.

Here are 20 AI tools to future-proof yourself. Get more tools like this by signing up to Superh...



**ResearchHub Foundation** @ResearchHubF · Sep 17

If you are not using AI, you are **falling behind!**

# What do people actually mean by “AI”?

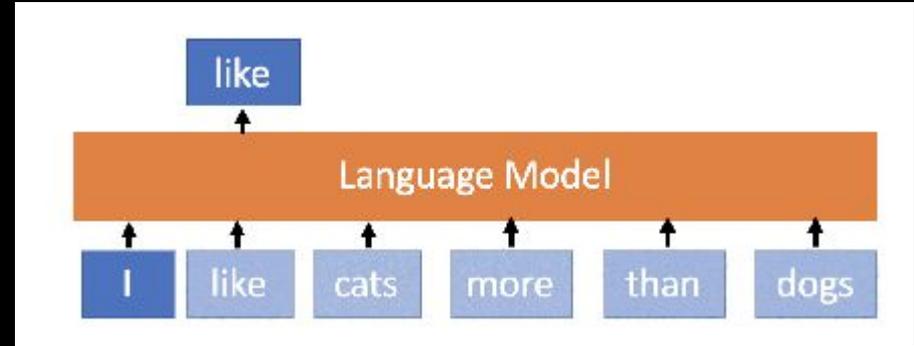
Large Language Models (Mostly)

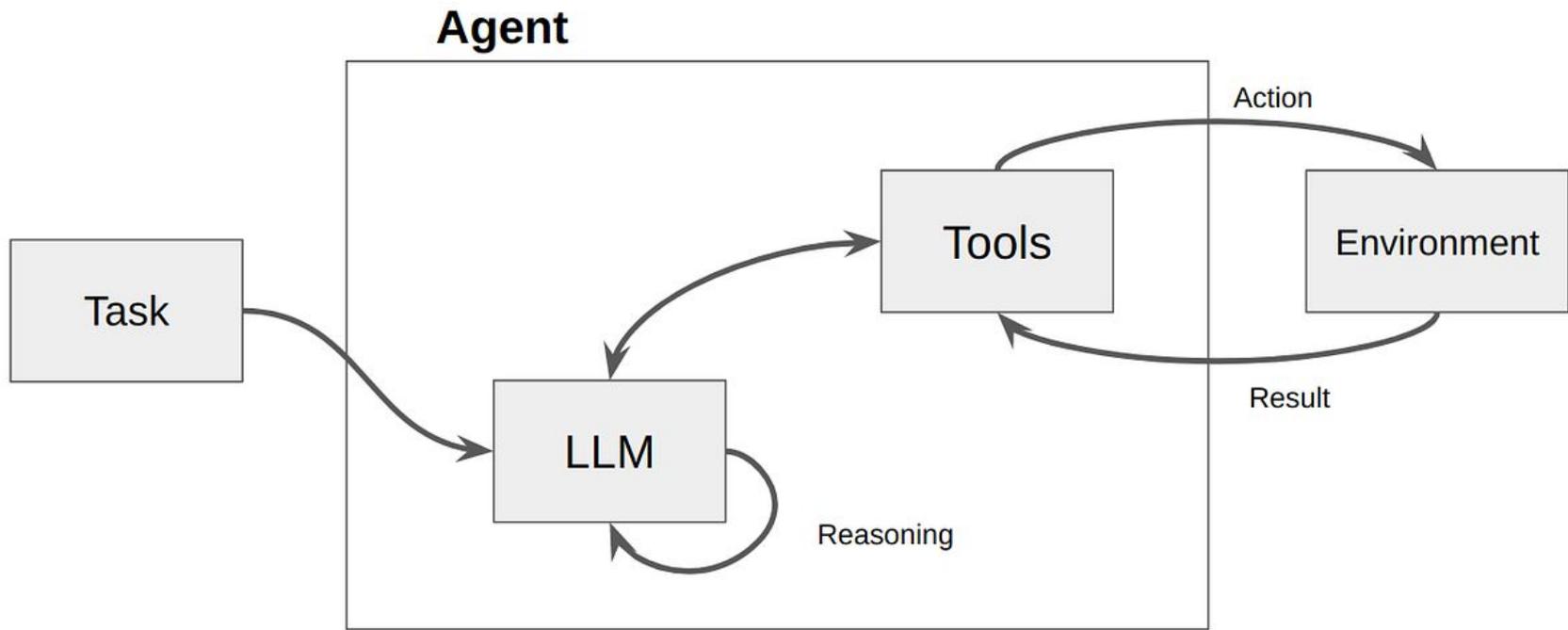
# Large Language Models

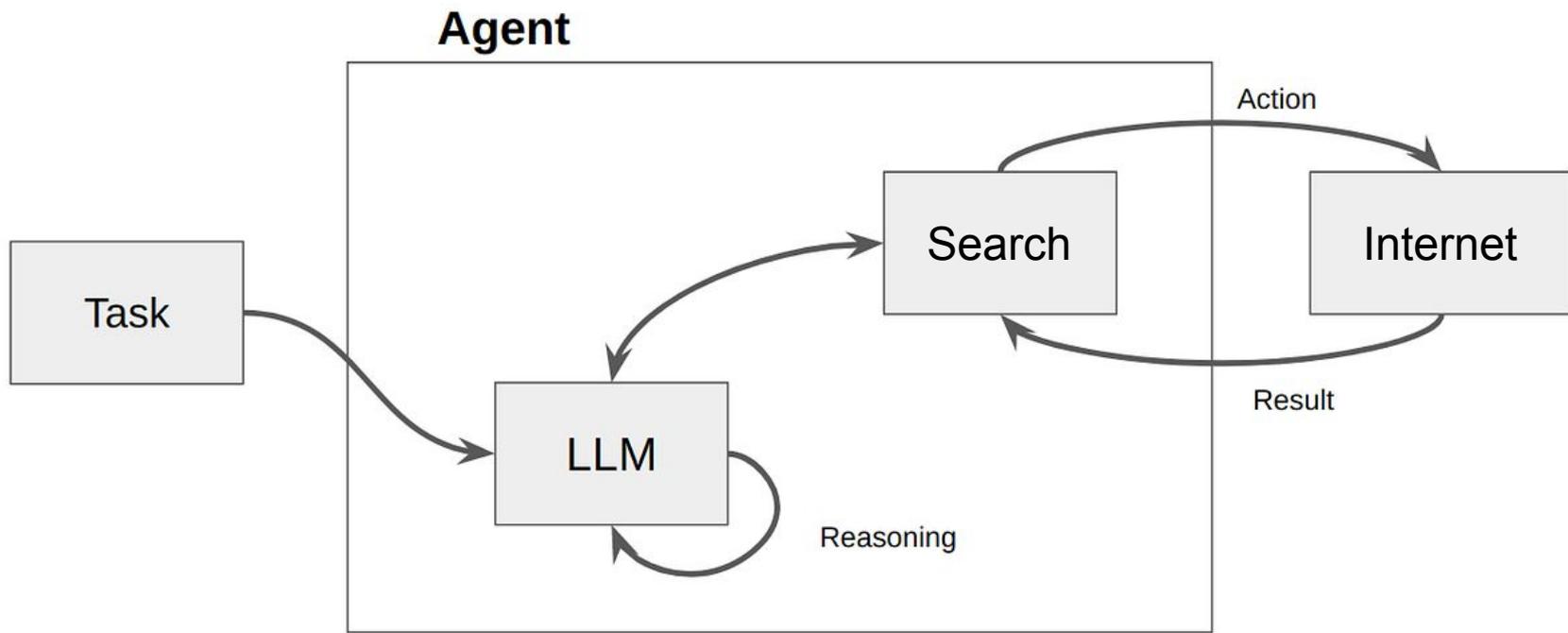
## ChatGPT

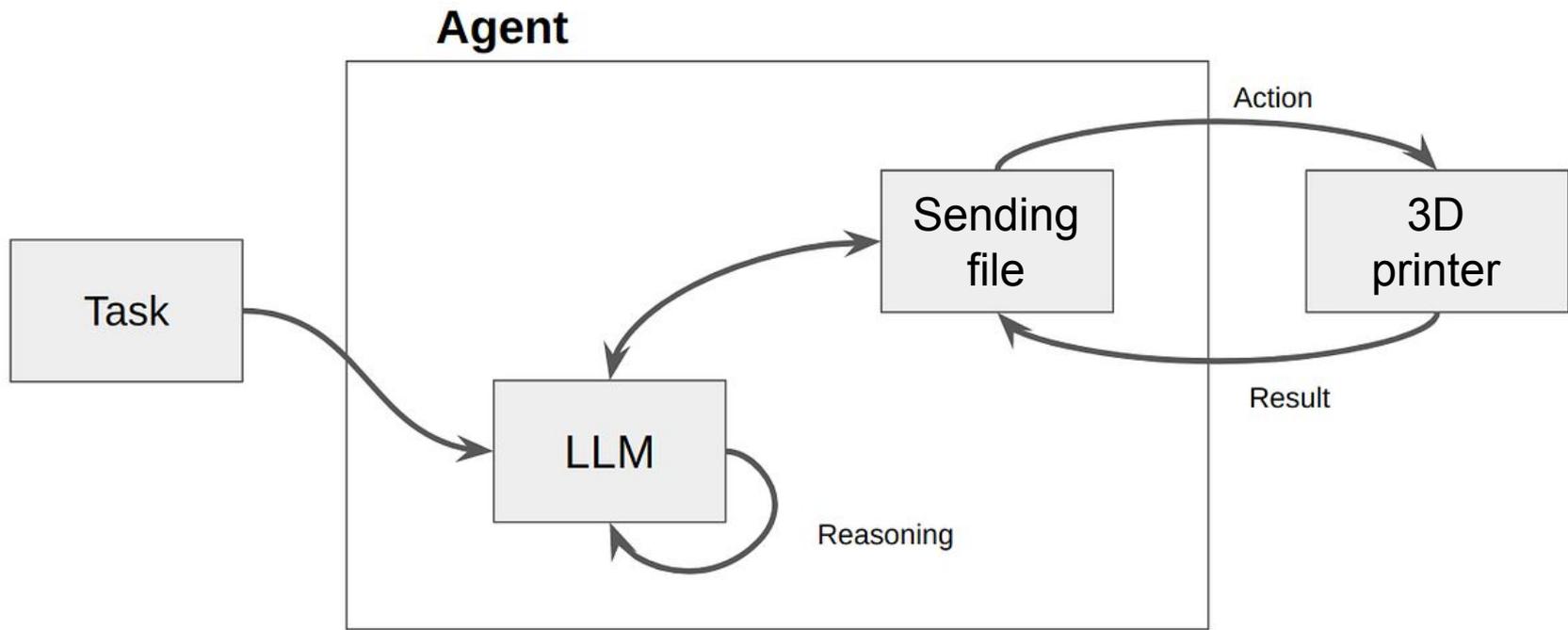
Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

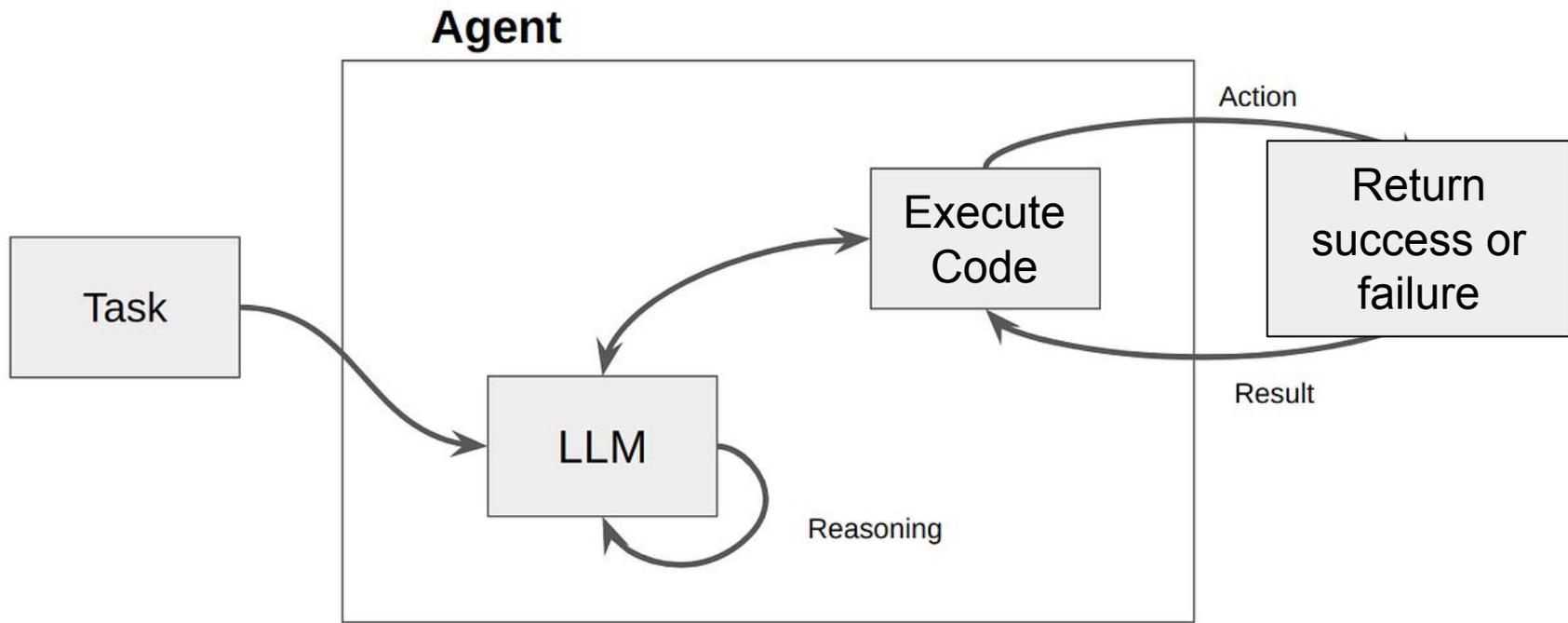
| ▾











# AgentGPT Beta

Assemble, configure, and deploy autonomous AI Agents in your browser.



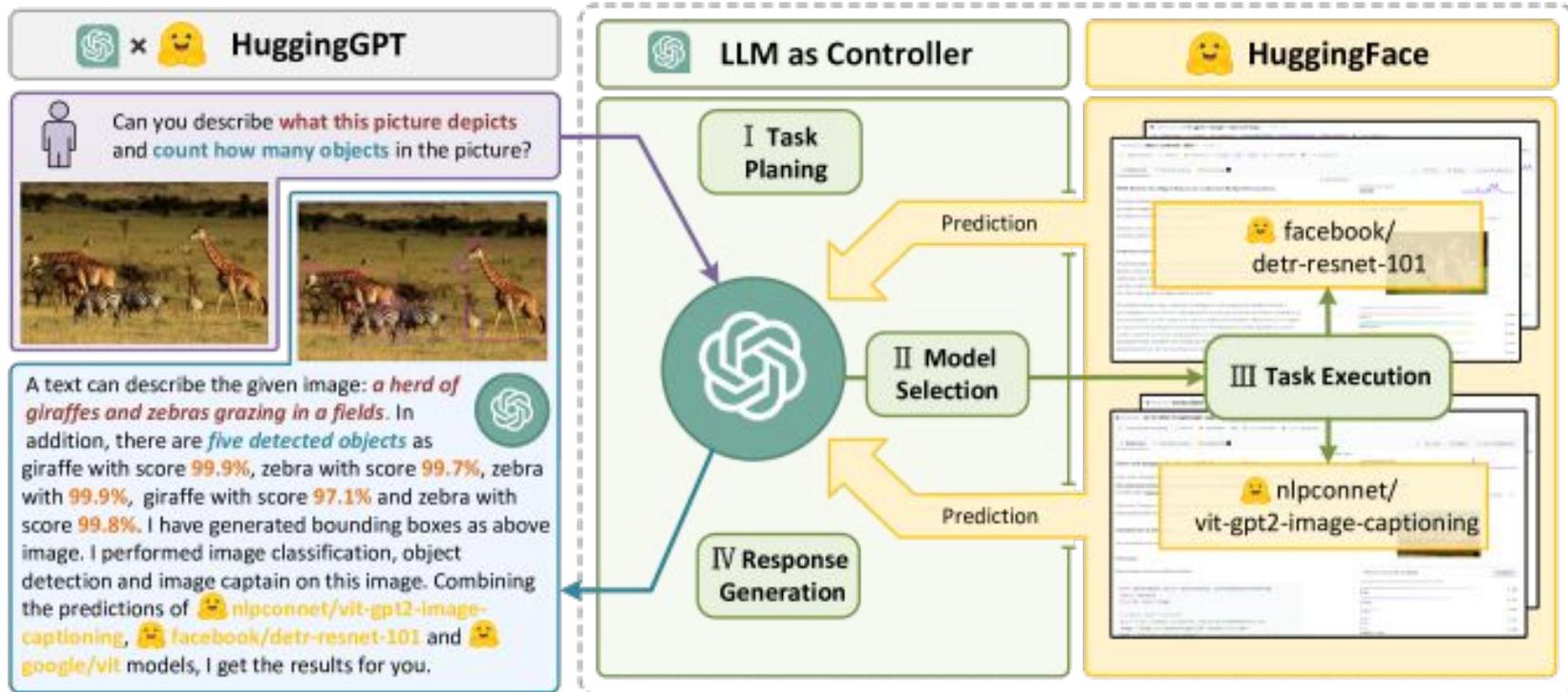
★ **Embarking on a new goal:** Create a new startup with only 100\$ of funding

🧠 **Thinking...**

📋 **Added task:** Research low-cost business models that require minimal funding.

📋 **Added task:** Identify potential target markets and their needs.

# Toolformers



New GPT

• Draft

Save

Create

Configure

Preview

**GPT Builder**

Hi! I'll help you build a new GPT. You can say something like, "make a creative who helps generate visuals for new products" or "make a software engineer who helps format my code."

What would you like to make?



Message GPT Builder...



Message GPT...





## Assistants API

[www.platform.openai.com](https://www.platform.openai.com)

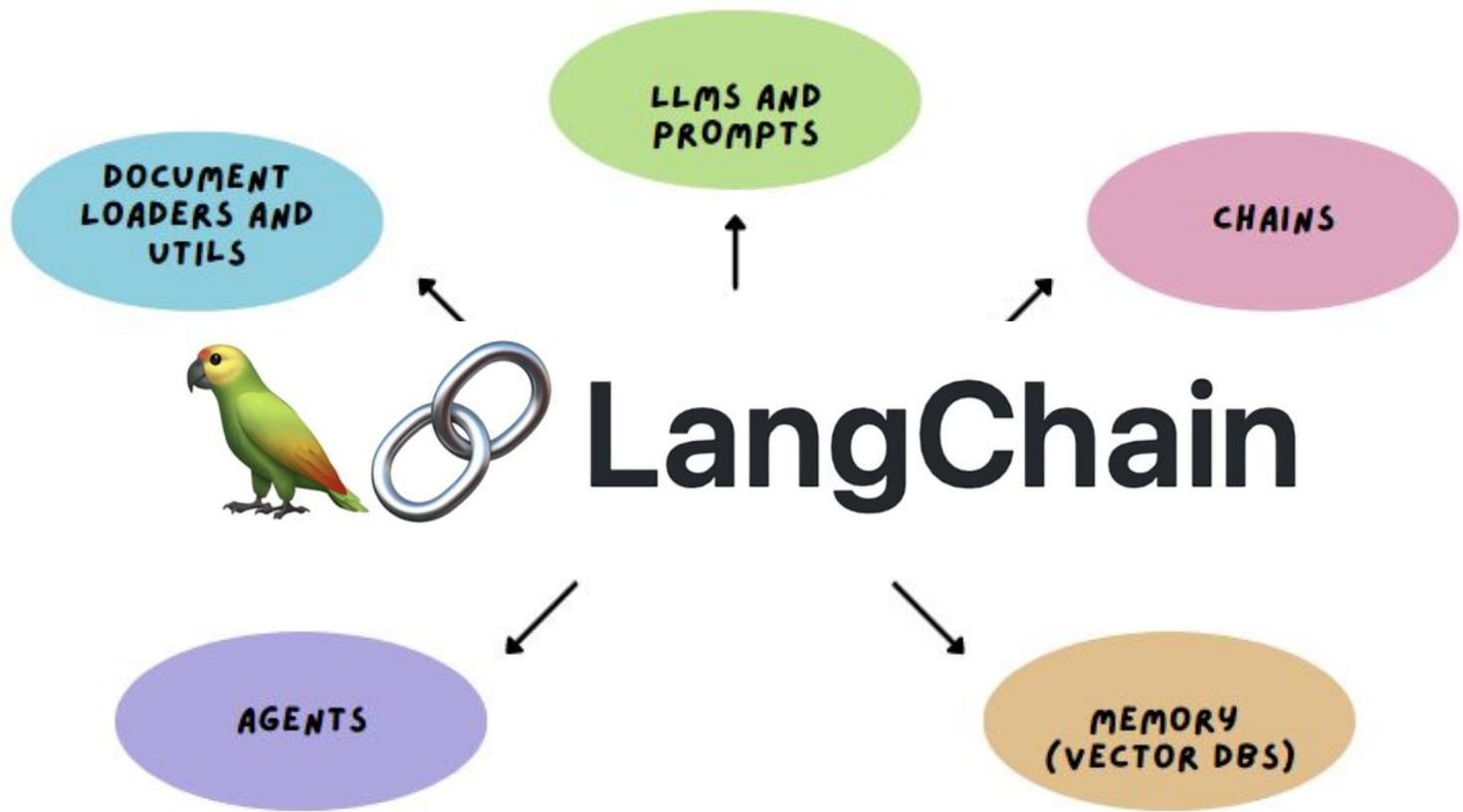
The screenshot shows the OpenAI Playground interface. On the left, the assistant is named "Galileo" and uses the "gpt-4-1106-preview" model. The instructions are: "You are a friendly assistant, your job is to help me answer questions about the universe." Under the "TOOLS" section, the "Code interpreter" tool is enabled. The right panel shows a thread with a user question: "How many days would it take to drive from Earth to the Moon if I were traveling at 60 miles per hour?". The code interpreter tool has executed the following Python code:

```
# Constants
distance_to_moon_miles = 238855 # Average distance from Earth to Moon in miles
speed_mph = 60 # Speed in miles per hour

# Calculate time in hours to reach the Moon
time_hours = distance_to_moon_miles / speed_mph

# Convert time to days
time_days = time_hours / 24
```

The output of the code interpreter is: 165.87152777777777. The assistant's response is: "It would take approximately 165.87 days to drive from Earth to the Moon at a constant speed of 60 miles per hour."





# Mistral 8x7b

Open-source

<https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1>

# GPT-4

Not open source

[platform.openai.com](https://platform.openai.com)



<https://github.com/cbh123/narrator>



# Hugging Face

## Multimodal

- Feature Extraction
- Text-to-Image
- Image-to-Text
- Text-to-Video
- Visual Question Answering
- Document Question Answering
- Graph Machine Learning

## Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Image-to-Image
- Unconditional Image Generation
- Video Classification
- Zero-Shot Image Classification

If you ever need something that isn't an LLM

## Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Conversational
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

## Audio

- Text-to-Speech
- Automatic Speech Recognition
- Audio-to-Audio
- Audio Classification
- Voice Activity Detection

## Tabular

- Tabular Classification
- Tabular Regression

## Reinforcement Learning

- Reinforcement Learning
- Robotics

<https://huggingface.co>

# Lets make each class (almost) obsolete!

## Computer-Aided Design and Manufacturing

- Generate CAD
- Laser cutting/Machining

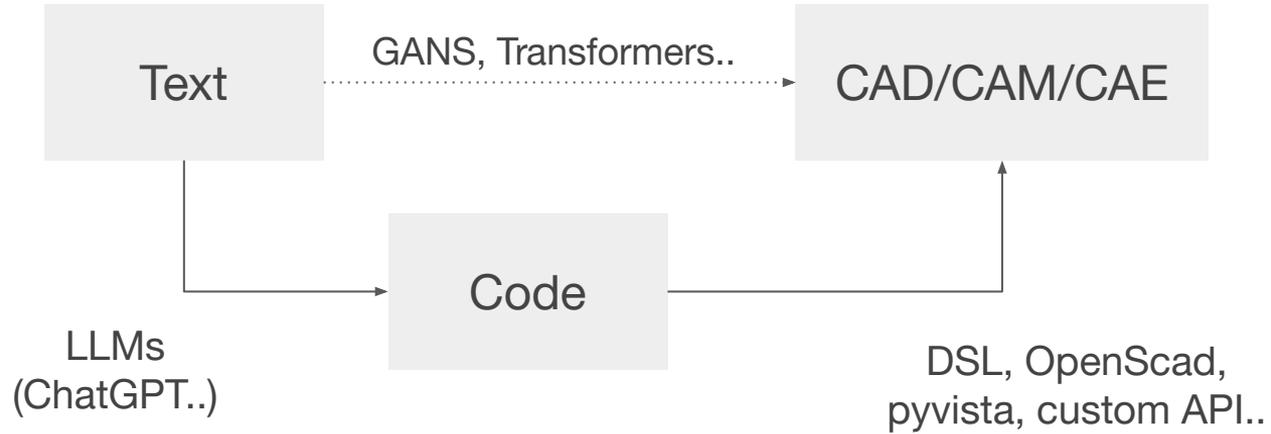
## 3D Modeling

- Text to Mesh

## Electronics Design and Programming

- Generate code
- Board Design

# **Computer-Aided Design and Manufacturing**



# CAD - using a Domain Specific Language (DSL)



Given a function `box(x, y, z, w, h, d)` which generates a 3D box of dimensions  $w \times h \times d$  centered at the 3D coordinates  $(x, y, z)$ , can you write a program to produce a simple box of size  $100 \times 100 \times 400$  utilizing the function `box` - [Link for full prompt](#)

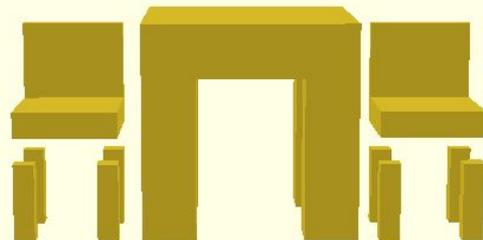
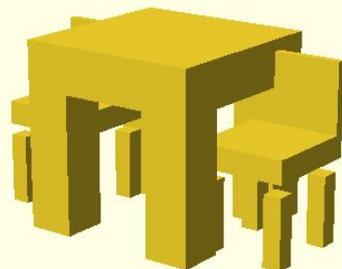
```
box(0, 0, 0, 100, 100, 400)
```



using the function `box` can you generate a table that has four legs and a tabletop of size  $400 \times 400 \times 100$ , and add a few accompanying chairs.

```
def create_table():  
    # Tabletop  
    box(0, 50, 0, 400, 100, 400)  
  
    # Four legs  
    # Bottom-left leg  
    box(-150, -150, -150, 100, 300, 100)  
    # Bottom-right leg  
    box(150, -150, -150, 100, 300, 100)  
    # Top-left leg  
    box(-150, -150, 150, 100, 300, 100)  
    # Top-right leg  
    box(150, -150, 150, 100, 300, 100)
```

```
create_table()
```

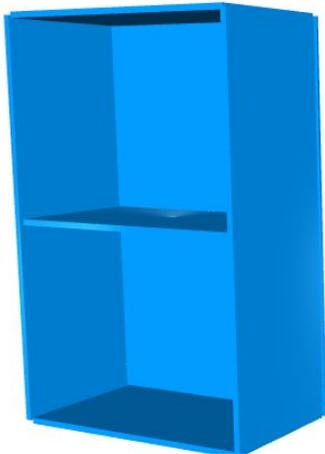


# CAD - OpenJSCAD



Design a simple cabinet with one shelf using openscad.

- [Link for full prompt](#)



<https://chat.openai.com/share/1fc8a8db-8776-4ab5-91fe-7a90c2c8c6fa>

Press 'shift + enter' to render!

```
const { cuboid } = require('@jscad/modeling').primitives;
const { union } = require('@jscad/modeling').booleans;
const { translate } = require('@jscad/modeling').transforms;

function main() {
  const boardThickness = 18; // Thickness of the cabinet board
  const cabinetWidth = 800; // Width of the cabinet
  const cabinetDepth = 400; // Depth of the cabinet
  const cabinetHeight = 1200; // Height of the cabinet
  const backBoardThickness = 9; // Thickness of the back board
  const shelfFromBottom = 600; // Shelf position from the bottom

  const halfWidth = cabinetWidth / 2;
  const halfDepth = cabinetDepth / 2;
  const halfHeight = cabinetHeight / 2;

  // Sides
  const sideWidth = boardThickness;
  const sideHeight = cabinetHeight;
  const sideDepth = cabinetDepth;
  const leftSide = translate([-halfWidth, 0, 0], cuboid({ size: [sideWidth, sideDepth, sideHeight] }));
  const rightSide = translate([halfWidth - boardThickness, 0, 0], cuboid({ size: [sideWidth, sideDepth, sideHeight] }));

  // Top & Bottom
  const top = translate([0, 0, halfHeight], cuboid({ size: [cabinetWidth - 2 * boardThickness, cabinetDepth, boardThickness] }));
  const bottom = translate([0, 0, -halfHeight], cuboid({ size: [cabinetWidth - 2 * boardThickness, cabinetDepth, boardThickness] }));

  // Back
  const back = translate([0, halfDepth - backBoardThickness / 2, 0], cuboid({ size: [cabinetWidth, backBoardThickness, cabinetHeight] }));

  // Shelf
  const shelf = translate([0, 0, shelfFromBottom - halfHeight], cuboid({ size: [cabinetWidth - 2 * boardThickness, cabinetDepth - boardThickness, boardThickness] }));

  return union(leftSide, rightSide, top, bottom, back, shelf);
}

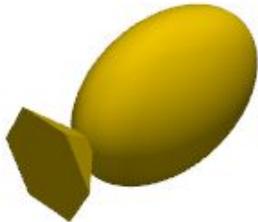
module.exports = { main };
```

# CAD - Python's PyVista



Use PyVista's primitives to model several variations of a fish, including specific bio-inspirations such as goldfish, a manta ray, and a loach - [Link for full prompt](#)

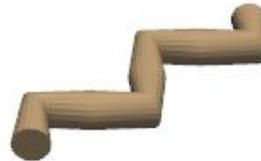
Goldfish



Manta Ray



Loach



# CAD - [text-to-cad.zoo.dev](https://zoo.dev/text-to-cad)

The screenshot displays the 'ZOO' interface for 'Text-to-CAD'. On the left, a sidebar lists prompts under 'Today' and 'Past 7 Days', each with a checkmark. The main area features a large text input field with a cursor, a 'SUBMIT' button, and a 'GPT-4' icon. Below the input field, 'EXAMPLE PROMPTS' are shown in a grid: '5-sided star', 'a plate with 4 holes near each corner and rounded corners', 'a 1/2 inch gear with 21 teeth', and 'a 9 pointed star'. A 'PROMPT WRITING TIPS' section follows, providing guidelines for writing prompts. At the bottom, there is a footer with 'Built with the DL-eshant API by Zoo' and 'View and contribute on GitHub'.

**ZOO**

**Today**

- Design a mounting plate that is 10... ✓
- Design a shelf bracket that can withstand 10... ✓
- Design a 1-inch diameter pipe that's ... ✓
- Design a 7 pointed star ✓
- Design a brick that's 10 inches wide, 6... ✓
- Design a mounting plate that's 5 inches... ✓

**Past 7 Days**

- Create a mounting plate that is 10... ✓
- Create a 22-tooth gear that has a module of... ✓

**Text-to-CAD**

e.g. Create a plate with 4 holes and rounded corners

I [GPT-4] [SUBMIT →]

**EXAMPLE PROMPTS:**

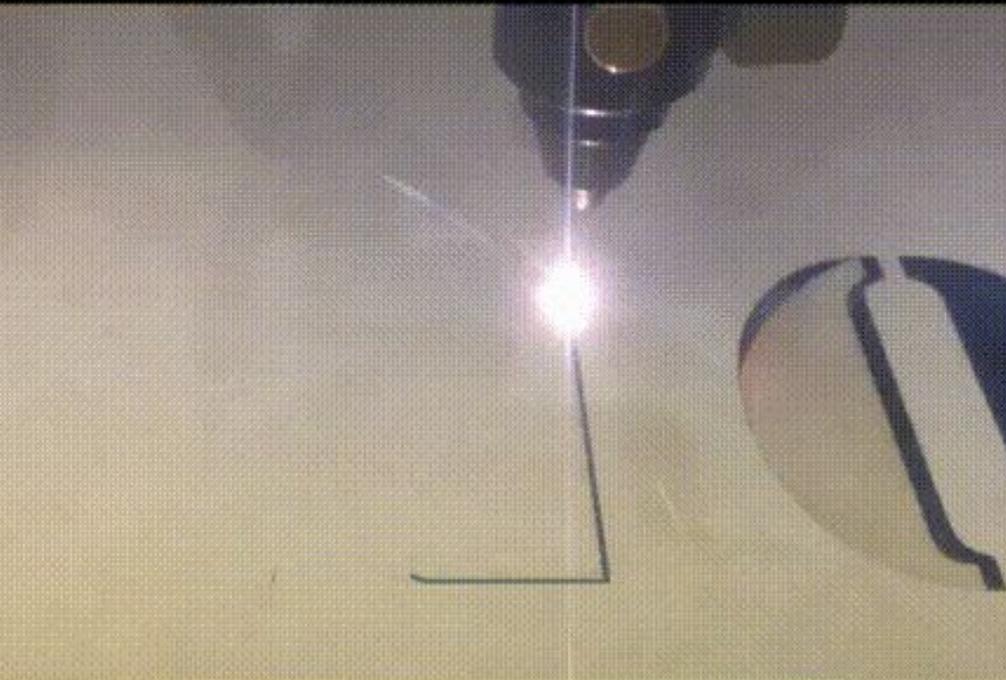
- 5-sided star
- a plate with 4 holes near each corner and rounded corners
- a 1/2 inch gear with 21 teeth
- a 9 pointed star

**PROMPT WRITING TIPS:**

- Describe an object that can be represented in geometric shapes, not nebulous concepts such as "a tiger" or "the universe", unless you're just curious what it does with that 🤖
- Be as explicit as possible. For example, if you want a plate with 4 holes, say where you want the holes placed and how big of a diameter each should have
- Our ML models are better at describing single objects than assemblies for now

Built with the DL-eshant API by Zoo | View and contribute on GitHub

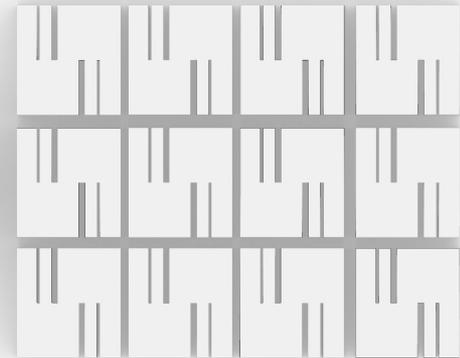
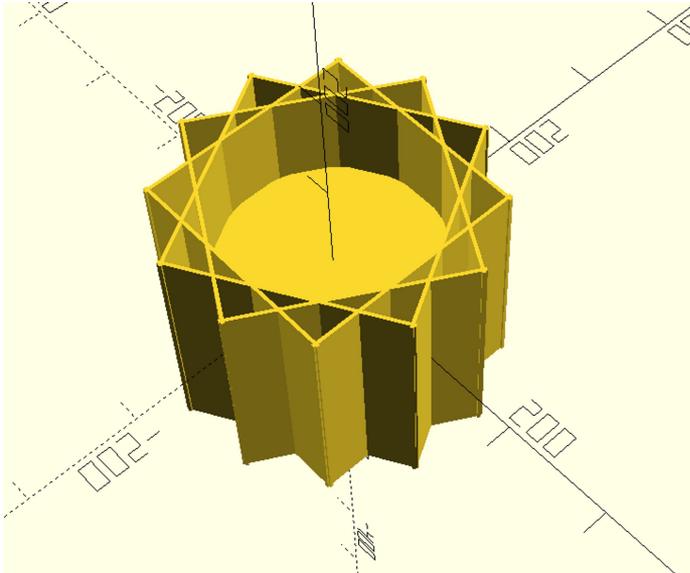
# CAD - Laser Cutting



# CAD - Laser Cutting



Design a multi-component geometric lamp that demonstrates the precision and intricacy achievable with laser cutting. Provide code to visualize the final lamp design using OpenSCAD.



# CAD - CNC



I want to fabricate a table with a cnc machine. I have wood stock with thickness 1.5 cm. Can you provide python code to create a svg file to cut the table top with press fit joints, and the legs with corresponding press fit joints.

```
import svgwrite

# Define the dimensions of the table components
thickness = 1.5 # cm
length = 120 # cm
width = 80 # cm
leg_height = 70 # cm
leg_width = 10 # cm
leg_thickness = 1.5 # cm
slot_width = 2 # cm
slot_length = 5 # cm
slot_depth = 2 # cm

# Create an SVG drawing
dwg = svgwrite.Drawing('table.svg', profile='tiny', size=(f'{length}cm', f'{width}cm'))

# Draw the table top
dwg.add(dwg.rect(insert=(0, 0), size=(f'{length}cm', f'{width}cm'), fill='none',
stroke=svgwrite.rgb(0, 0, 0, '%')) # No fill, black stroke

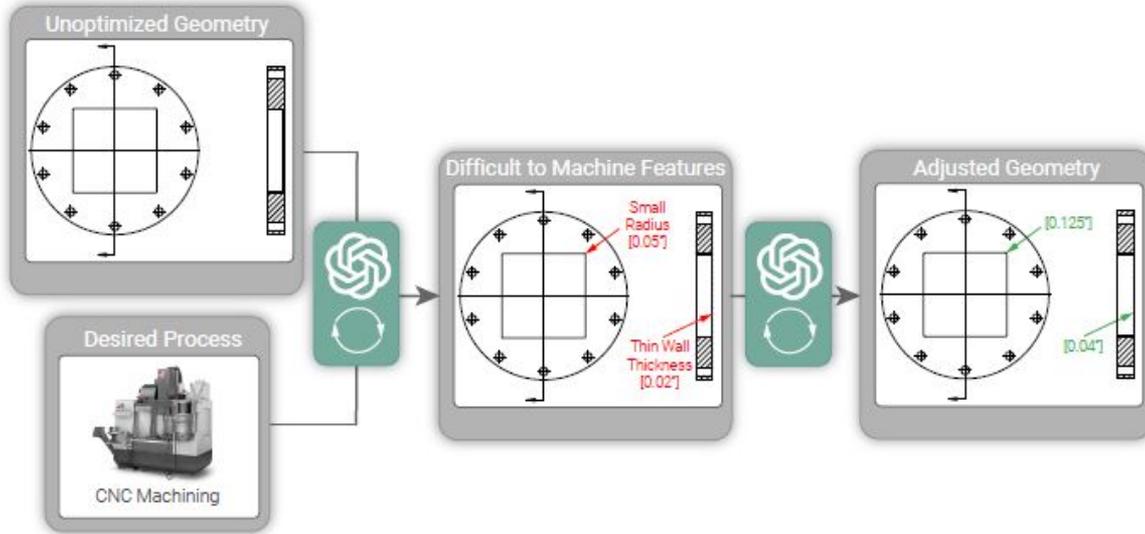
# Draw the legs and corresponding slots
for x in [0, length - leg_width]:
    for y in [0, width - leg_width]:
        # Draw the leg
        dwg.add(dwg.rect(insert=(f'{x}cm', f'{y}cm'), size=(f'{leg_width}cm', f'{leg_height}cm'), fill='none', stroke=svgwrite.rgb(0, 0, 0, '%')) # No fill, black stroke

        # Draw the slot on the table top
        slot_x = x + (leg_width - slot_width) / 2
        slot_y = y + (leg_width - slot_depth) / 2
        dwg.add(dwg.rect(insert=(f'{slot_x}cm', f'{slot_y}cm'), size=(f'{slot_length}cm', f'{slot_width}cm'), fill='none', stroke=svgwrite.rgb(0, 0, 0, '%')) # No fill, black stroke

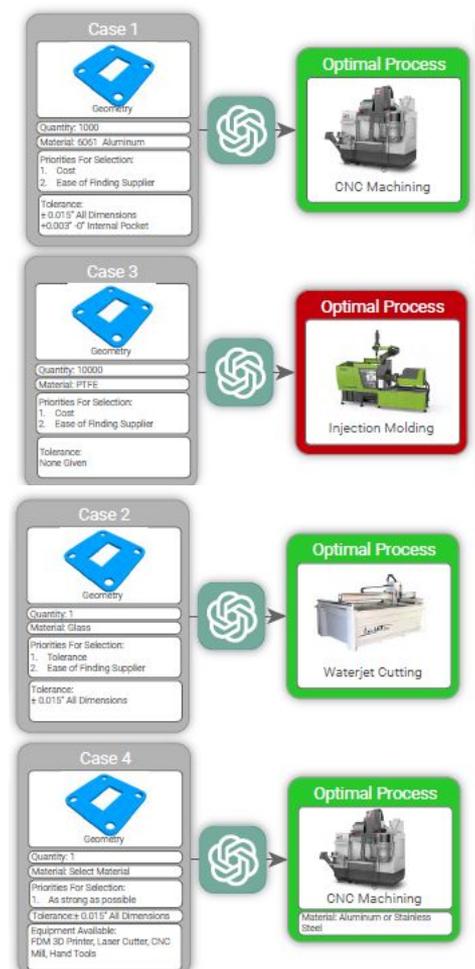
# Save the SVG file
dwg.save()
```



# LLMs and Computer Controlled Machining



Detect and Correct Manufacturing Problems [1]



Select Optimal Manufacturing Process [1]

# 3D Printing



The object I want to print is a simple solid 3D ball with radius of 1. Generate a sample g-code with a simple slicing for it in a virtual printer - [Prompt Link](#)



Can you generate G-code to 3d print a gyroid that is 5cm by 5cm by 5cm

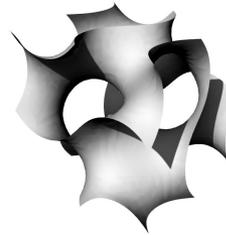
G-code generation is complex and it is heavily dependent on the specifics of the 3D printer being used, the material properties, the specific layer thickness, the nozzle size, print speed, and more. Hence, it's a process that should be done using a slicing software.

## Solution:

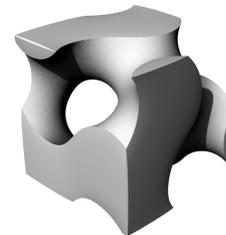
- I. Ask ChatGPT to turn the geometry into an intermediate 3D shape representation using triangle meshes (using trimesh in python).
- II. Translate this intermediate representation into G-Code using slic3r (python call) (visualize the output G-Code using Repetier).

- For simplicity's sake, we'll consider:
  - The printer has a 0.4mm nozzle.
  - Layer height is 0.2mm.
  - We're starting from the printer's home position.
  - No temperature settings or bed leveling is included, this is purely the movements

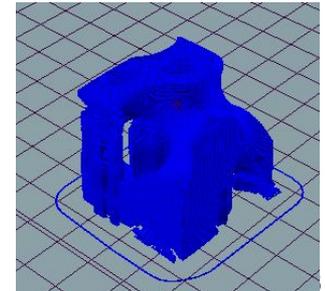
```
; Simplified G-code to print a 1cm sphere
; WARNING: This is a sample and may not be fit for actual
printing without adjustments
G21 ; Set units to millimeters
G90 ; Absolute positioning
; Initialize and home
G28 ; Home all axes
; Let's assume some simple print starting conditions:
G92 E0 ; Zero the extruder
G1 F1400 ; Set feed rate
; Start the print; this will only draw the base circle for the
first layer
G1 Z0.2 ; Move to first layer height
G1 X0 Y-10 ; Starting point
G1 E5 ; Prime the extruder
```



Open Mesh



Corrected Closed Mesh



G-code Generated by Slic3r

# Computer Controlled Machining - Performance Analysis

Given this chair made of wood, that has 4 legs that are 1 meter long (... content omitted by authors ...) can it support 100kg? Please give a final yes or no answer.

To determine whether the chair can support 100kg, we need to consider the strength of the wood used to construct the chair, as it determines the load-bearing capacity. However, this information was not provided. Nonetheless, let's assume that the wood used is Oak, which has a fairly high compressive strength of around 37 MPa or 37,000,000 Pascals. (... content omitted by authors ...) If we consider each leg bears 1/4 of the total weight, each leg will support 25kg. The pressure (force per unit area) on each leg then is  $\text{Force/Area} = (25\text{kg} * 9.8 \text{ m/s}^2) / 0.00258064 \text{ m}^2 = 94,692.2 \text{ Pascals}$  or 94.7 kPa, which is well within the compressive strength of Oak, so the legs should hold up. (... content omitted by authors ...) Thus, based on these calculations and assumptions about the distribution of weight and the type of wood used, the answer is Yes, the chair can support 100kg.

Give me a function that can evaluate whether a chair of a typical design will break under a certain applied weight. Please consider multiple types of failure.

```
def will_chair_break (weight, leg_cross_sectional_area, leg_material_strength, seat_thickness,
                    seat_length, seat_width, seat_material_bending_strength, back_height, back_width,
                    back_material_strength):
    # Calculate the load on each leg
    load_per_leg = weight / 4
    # Calculate the stress on each leg
    stress_on_leg = load_per_leg / leg_cross_sectional_area
    # Check if the stress on each leg exceeds the strength of the leg material
    if stress_on_leg > leg_material_strength:
        return True # The chair will break
    (... omitted by authors: code continues on to assess seat and back bending ...)
```

Analytical Verification [1]

Properties assessed



Give me one open-source FEM library.

Sure, one open-source FEM (Finite Element Method) library is "FEniCS".

Can you please use Python code to implement FEM linear elasticity to determine if the chair will break? i.e., if the stress will violate the yield constraint? Assume the bottom is fixed and a constant downward load is applied at the top. You can use FEniCS. The problem is in 3D. You can assume a simple mesh.

```
from fenics import *
from ufl import nabla_div
V = VectorFunctionSpace(mesh, 'P', 1)

# Define boundary condition
(... omitted by authors: boundary details ...)
bc = DirichletBC(V, Constant((0, 0, 0)), boundary_bottom)

# Define strain and stress
def epsilon(u):
    return 0.5*(nabla_grad(u) + nabla_grad(u).T)
def sigma(u):
    (... omitted by authors: modulus definition ...)
    mu = 1 # shear modulus
    lmbda = 1 # first Lamé's parameter
    return lmbda*nabla_div(u)*Identity(3) + 2*mu*epsilon(u)

# Define variational problem
u = TrialFunction(V)
d = TestFunction(V)
f = Constant((0, 0, -10)) # body force
T = Constant((0, 0, 0)) # traction
a = inner(sigma(u), epsilon(d))*dx
L = dot(f, d)*dx + dot(T, d)*ds

# Compute solution
u = Function(V)
solve(a == L, u, bc)

# Compute Von Mises Stress
s = sigma(u) - (1./3)*tr(sigma(u))*Identity(3) # 3 for 3D
von_Mises = abs(sqrt(3./2*inner(s, s)))
(... omitted by authors: von_Mises details, setting the yield stress ...)

# Check if the yield stress is violated
yield_violation = VonMises.vector().get_local() > yield_stress
if any(yield_violation):
    print("Yield constraint is violated.")
else:
    print("Yield constraint is not violated.")
(... omitted by authors: saving results to file ...)
```

FEM [1]



# CAD - Strengths and Limitations

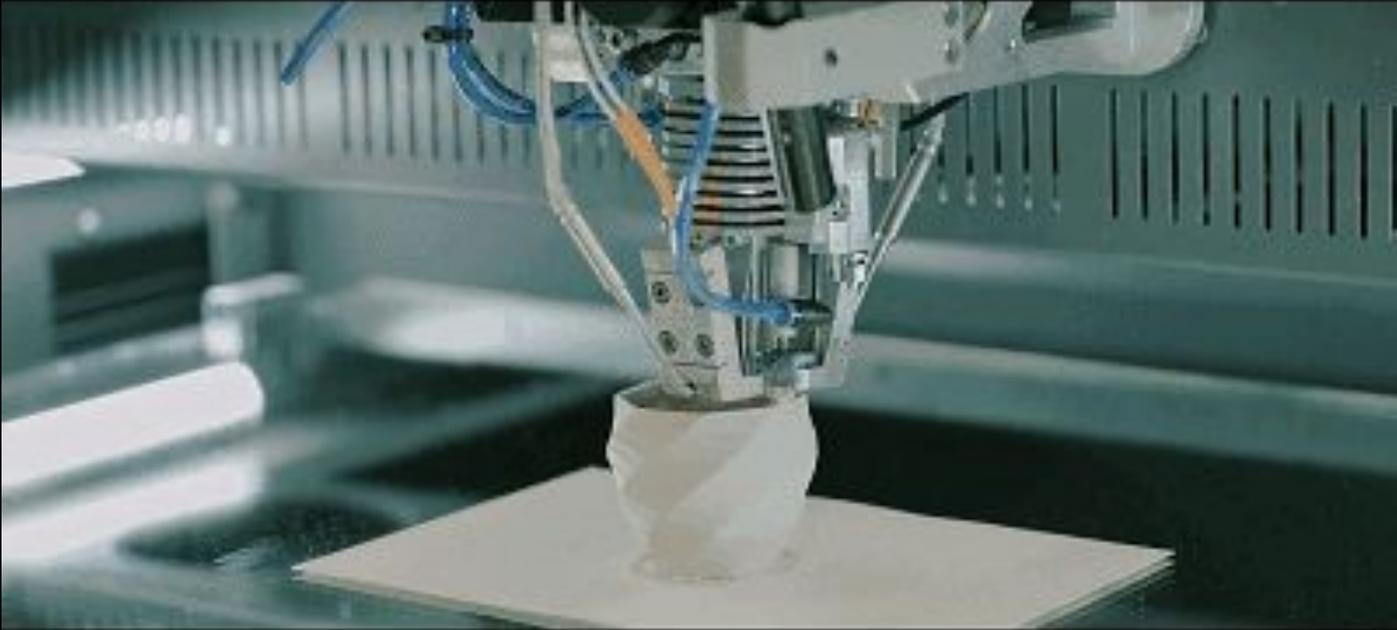
## Strengths:

- Respects **high-level spatial constraints**, such as a design element's absolute size or its position relative to another element of the design.
- **Iteration Support**
- **Parametric design**: ability to create parameters, bounds and constraints for text-based designs and already existing, and interpolate and change designs
- **Modularity and Hierarchy**: works better when submodules are formed first.

## Limitations:

- **Lack of spatial awareness** created difficulties with constraint handling
  - Fails to respect “non-overlapping” constraints, but it generally responds well to the instruction that some element should be “in contact with (but not protruding into)” another element
- **Problem with Scalability**:
  - Short attention span
  - Success seems to decline as the number of simultaneous requests increases (it is best to issue 1-2 constraints or correct 1-2 issues)
- **Inability to verify** its own output

# 3D Scanning & Modeling



# The (tough) old way

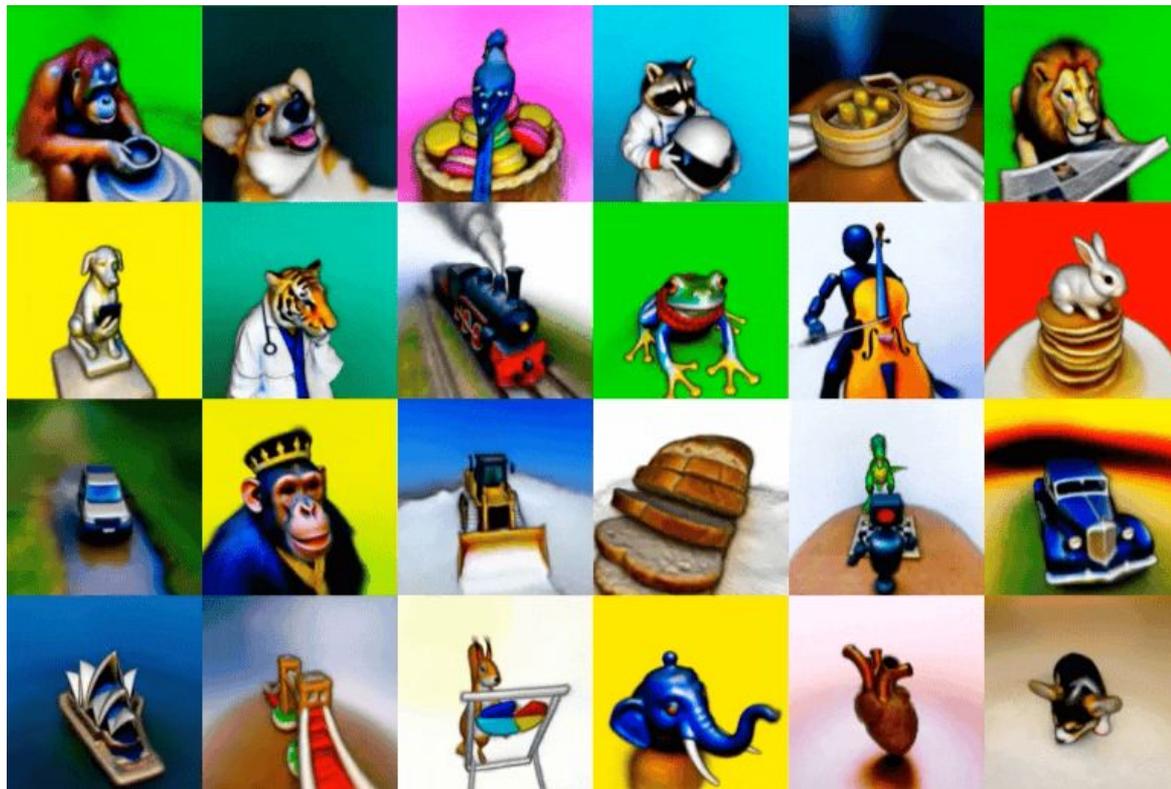


# Dreamfields (May 2022)

CLIP → NeRF



# Dreamfusion (Sept 2022)



[Stable Dreamfusion Code](#)

# Magic3D (Nov 2022)

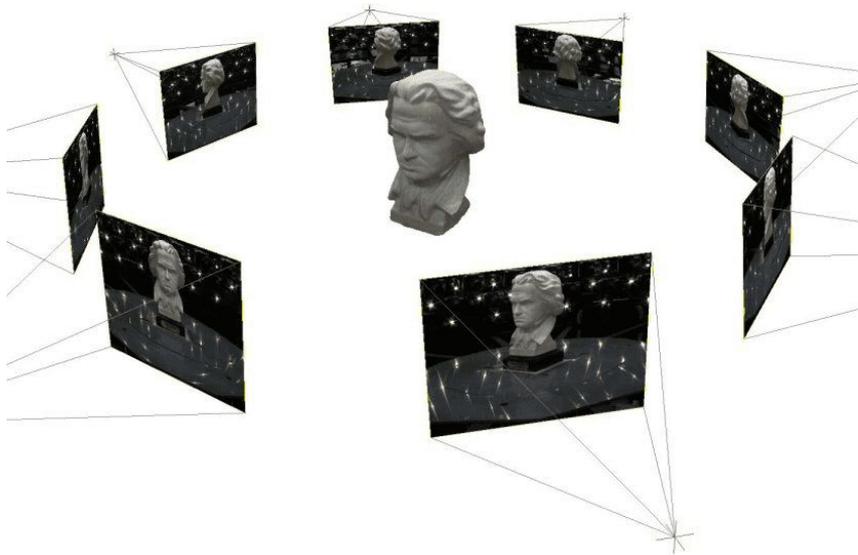
Michelangelo style statue  
of an astronaut.



DreamFusion



Magic3D



Images to 3D model

“Teapot  
made  
out of  
leaves”

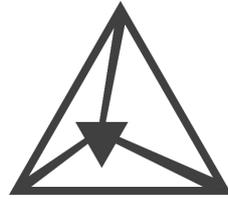


Text to 3D model



LUMA AI

<https://github.com/nerfstudio-project/nerfstudio>



# threestudio



<https://github.com/threestudio-project/threestudio>



GENIE

Imagine

Creations



<https://lumalabs.ai/genie>

monkey riding a giant sausage

Create ✨



Valdemar Danry x Cenk Güzelis

PRESENTS

# ORGANs WITHOUT BODIES

OPENING: \_5PM, JULY 17  
MIT MEDIA LAB, E14 LOBBY





“Salt shaker in the shape of an Axolotl”



“Teapot made out of leaves”

**Try it yourself:**

**<https://github.com/valleballe/depthfusion/>**

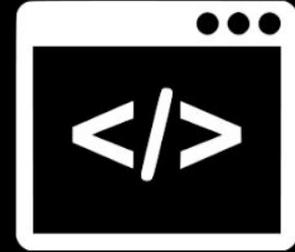
# Electronics Design



1. Specifications



2. Schematics



3. Programming

# 1. Spe

## FD1771-01 Floppy Disk Formatter/Controller

### FEATURES

- **SOFT SECTOR FORMAT COMPATIBILITY**
- **AUTOMATIC TRACK SEEK WITH VERIFICATION**
- **READ MODE**
  - Single/Multiple Sector Write with Automatic Sector Search or Entire Track Read
  - Selectable 128 Byte or Variable Length Sector
- **WRITE MODE**
  - Single/Multiple Sector Write with Automatic Sector Search
  - Entire Track Write for Diskette Formatting
- **PROGRAMMABLE CONTROLS**
  - Selectable Track-to-Track Stepping Time
  - Selectable Head Setting and Head Engage Times
  - Selectable Three Phase or Step and Direction and Head Positioning Motor Controls

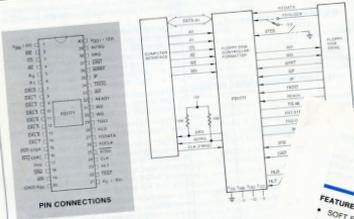
- **SYSTEM COMPATIBILITY**
  - Double Buffering of Data 8-Bit Bi-Directional Bus for Data, Control and Status
  - DMA or Programmed Data Transfers
  - All Inputs and Outputs are TTL Compatible

### APPLICATIONS

- FLOPPY DISK DRIVE INTERFACE
- SINGLE OR MULTIPLE DRIVE CONTROLLER/FORMATTER
- NEW MINI-FLOPPY CONTROLLER

### GENERAL DESCRIPTION

The FD1771 is a MOS/LSI device that performs the functions of a Floppy Disk Controller/Formatter. The device is designed to be included in the disk drive electronics, and contains a flexible interface

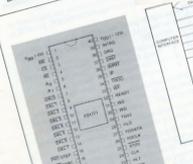


FD1771 SYSTEM BLOCK DIAGRAM

## FD1771-01 Floppy Di

### FEATURES

- **SOFT SECTOR FORMAT COMPATIBILITY**
- **AUTOMATIC TRACK SEEK WITH VERIFICATION**
- **READ MODE**
  - Single/Multiple Sector Write with Automatic Sector Search or Entire Track Read
  - Selectable 128 Byte or Variable Length Sector
- **WRITE MODE**
  - Single/Multiple Sector Write with Automatic Sector Search
  - Entire Track Write for Diskette Formatting
- **PROGRAMMABLE CONTROLS**
  - Selectable Track-to-Track Stepping Time
  - Selectable Head Setting and Head Engage Times
  - Selectable Three Phase or Step and Direction and Head Positioning Motor Controls



FD1771 SYSTEM BLOCK DIAGRAM

## FD1771-01 Floppy Disk Formatter/Controller

### FEATURES

- **SOFT SECTOR FORMAT COMPATIBILITY**
- **AUTOMATIC TRACK SEEK WITH VERIFICATION**
- **READ MODE**
  - Single/Multiple Sector Write with Automatic Sector Search or Entire Track Read
  - Selectable 128 Byte or Variable Length Sector
- **WRITE MODE**
  - Single/Multiple Sector Write with Automatic Sector Search
  - Entire Track Write for Diskette Formatting
- **PROGRAMMABLE CONTROLS**
  - Selectable Track-to-Track Stepping Time
  - Selectable Head Setting and Head Engage Times
  - Selectable Three Phase or Step and Direction and Head Positioning Motor Controls

### SYSTEM COMPATIBILITY

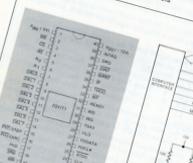
- Double Buffering of Data 8-Bit Bi-Directional Bus for Data, Control and Status
- DMA or Programmed Data Transfers
- All Inputs and Outputs are TTL Compatible

### APPLICATIONS

- FLOPPY DISK DRIVE INTERFACE
- SINGLE OR MULTIPLE DRIVE CONTROLLER/FORMATTER
- NEW MINI-FLOPPY CONTROLLER

### GENERAL DESCRIPTION

The FD1771 is a MOS/LSI device that performs the functions of a Floppy Disk Controller/Formatter. The device is designed to be included in the disk drive electronics, and contains a flexible interface



FD1771 SYSTEM BLOCK DIAGRAM

## FD1771-01 Floppy Disk Formatter/Controller

### FEATURES

- **SOFT SECTOR FORMAT COMPATIBILITY**
- **AUTOMATIC TRACK SEEK WITH VERIFICATION**
- **READ MODE**
  - Single/Multiple Sector Write with Automatic Sector Search or Entire Track Read
  - Selectable 128 Byte or Variable Length Sector
- **WRITE MODE**
  - Single/Multiple Sector Write with Automatic Sector Search
  - Entire Track Write for Diskette Formatting
- **PROGRAMMABLE CONTROLS**
  - Selectable Track-to-Track Stepping Time
  - Selectable Head Setting and Head Engage Times
  - Selectable Three Phase or Step and Direction and Head Positioning Motor Controls

### SYSTEM COMPATIBILITY

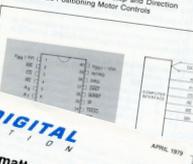
- Double Buffering of Data 8-Bit Bi-Directional Bus for Data, Control and Status
- DMA or Programmed Data Transfer
- All Inputs and Outputs are TTL Compatible

### APPLICATIONS

- FLOPPY DISK DRIVE INTERFACE
- SINGLE OR MULTIPLE DRIVE CONTROLLER/FORMATTER
- NEW MINI-FLOPPY CONTROLLER

### GENERAL DESCRIPTION

The FD1771 is a MOS/LSI device that performs the functions of a Floppy Disk Controller/Formatter. The device is designed to be included in the disk drive electronics, and contains a flexible interface



FD1771 SYSTEM BLOCK DIAGRAM

## FD1771-01 Floppy Disk Formatter/Controller

### FEATURES

- **SOFT SECTOR FORMAT COMPATIBILITY**
- **AUTOMATIC TRACK SEEK WITH VERIFICATION**
- **READ MODE**
  - Single/Multiple Sector Write with Automatic Sector Search or Entire Track Read
  - Selectable 128 Byte or Variable Length Sector
- **WRITE MODE**
  - Single/Multiple Sector Write with Automatic Sector Search
  - Entire Track Write for Diskette Formatting
- **PROGRAMMABLE CONTROLS**
  - Selectable Track-to-Track Stepping Time
  - Selectable Head Setting and Head Engage Times
  - Selectable Three Phase or Step and Direction and Head Positioning Motor Controls

### SYSTEM COMPATIBILITY

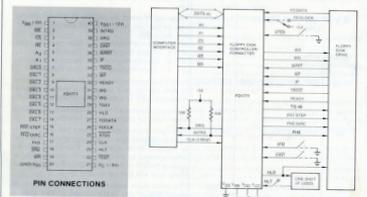
- Double Buffering of Data 8-Bit Bi-Directional Bus for Data, Control and Status
- DMA or Programmed Data Transfers
- All Inputs and Outputs are TTL Compatible

### APPLICATIONS

- FLOPPY DISK DRIVE INTERFACE
- SINGLE OR MULTIPLE DRIVE CONTROLLER/FORMATTER
- NEW MINI-FLOPPY CONTROLLER

### GENERAL DESCRIPTION

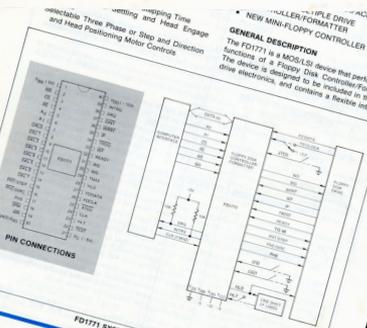
The FD1771 is a MOS/LSI device that performs the functions of a Floppy Disk Controller/Formatter. The device is designed to be included in the disk drive electronics, and contains a flexible interface



FD1771 SYSTEM BLOCK DIAGRAM

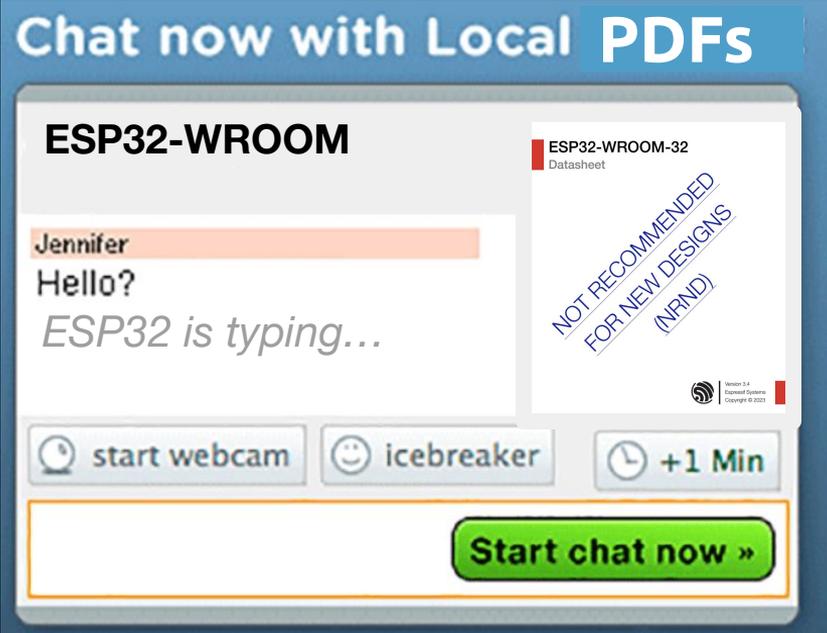
## Controller

- 8-Bit Bi-Directional Status
- Compatible



FD1771 SYSTEM BLOCK DIAGRAM

# 1. Specifications



The screenshot shows a chat window titled "Chat now with Local PDFs". The chat content includes a message from "Jennifer" saying "Hello?" and a status message "ESP32 is typing...". To the right, a PDF document titled "ESP32-WROOM-32 Datasheet" is displayed with a prominent diagonal warning: "NOT RECOMMENDED FOR NEW DESIGNS (NRND)". Below the chat area are three buttons: "start webcam", "icebreaker", and "+1 Min". At the bottom right is a large green button labeled "Start chat now »".

<https://chatpdf.com>

# Chat with any PDF

Join millions of students, researchers and professionals to instantly answer questions and understand research with AI



Drop PDF here

[Browse my Computer](#)

[From URL](#)

## Going Viral all over the World

Across borders, beyond languages: AI is revolutionizing the understanding of research worldwide

## 2. Schematics

OF

Can you tell me which pins to connect between an ESP32 and a LSM6DS33 accelerometer?



Certainly! The LSM6DS33 is a digital accelerometer and gyroscope sensor that communicates via I2C or SPI. In this answer, I'll provide the connections for both I2C and SPI.



### 1. I2C Connection:

On the ESP32:

- `GPIO21` is typically the default SDA (I2C data) pin.
- `GPIO22` is typically the default SCL (I2C clock) pin.

On the LSM6DS33:

- Connect `SDA` to ESP32's `GPIO21`.
- Connect `SCL` to ESP32's `GPIO22`.

Common Connections for both I2C and SPI:

- Connect `GND` (ground) of LSM6DS33 to `GND` of ESP32.
- Connect `VDD` of LSM6DS33 to `3.3V` of ESP32.

# 3. Embedded Programming

OF

Provide Arduino code to read magnetic field values from a TLE493D-A2B6



Using the TLE493D-A2B6 sensor with an Arduino requires an appropriate library, as the TLE493D-A2B6 communicates using I2C and has its own specific register and data format. As of my last training data (January 2022), Infineon provided a library for the TLE493D-A2B6.



Here's a basic Arduino sketch to read magnetic field values from a TLE493D-A2B6 using the provided library:

1. Install the library.
2. Wire the TLE493D-A2B6 to your Arduino.
3. Use the following code:

cpp

Copy code

```
#include <Wire.h>
#include <Tle493d_w2b6.h>

Tle493d_w2b6 Tle493dMagnetic3DSensor = Tle493d_w2b6();

void setup() {
  Serial.begin(9600);

  // Initialize I2C
  Wire.begin();
```

But if LLMs can do all of these steps, then why don't we just put it all up into one package?

<https://github.com/valleballe/BoardDesignerGPT/>

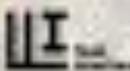
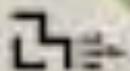
# BoardDesignerGPT

Simplifying electronics design with  
LLM-agents

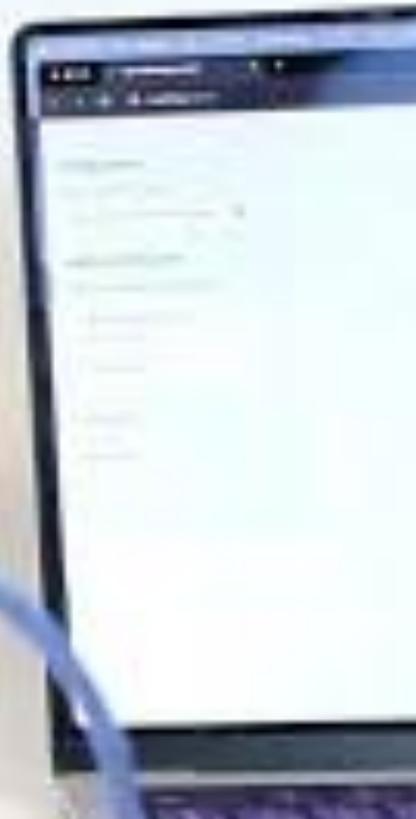
Valdemar Darry  
Pattie Maes  
Neil Gershenfeld



Massachusetts  
Institute of  
Technology



THE CENTER FOR  
BITS AND ATOMS  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY



**Examples of cool things  
people have made**

**Be My Eyes**  
Support for vision impaired



**RizzGPT**

Suggests you what to say

# Can an AI Student (Almost) Make Things?

**LLMs make it *easier* to write code / produce artifacts,  
but it is up to humans to *exhibit creativity and  
judgment.***

## Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park  
Stanford University  
Stanford, USA  
joonspk@stanford.edu

Joseph C. O'Brien  
Stanford University  
Stanford, USA  
jobrien3@stanford.edu

Carrie J. Cai  
Google Research  
Mountain View, CA, USA  
cjcai@google.com

Meredith Ringel Morris  
Google DeepMind  
Seattle, WA, USA  
merrie@google.com

Percy Liang  
Stanford University  
Stanford, USA  
pliang@cs.stanford.edu

Michael S. Bernstein  
Stanford University  
Stanford, USA  
mab@cs.stanford.edu

arXiv:2304.03442v2 [cs.HC] 6 Aug 2023



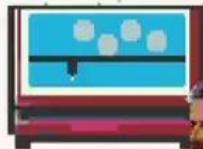
Figure 1: Generative agents are believable simulacra of human behavior for interactive applications. In this work, we demonstrate generative agents by populating a sandbox environment, reminiscent of *The Sims*, with twenty-five agents. Users can observe and intervene as agents plan their days, share news, form relationships, and coordinate group activities.

### ABSTRACT

Believable proxies of human behavior can empower interactive applications ranging from immersive environments to rehearsal spaces for interpersonal communication to prototyping tools. In this paper, we introduce generative agents: computational software agents that simulate believable human behavior. Generative agents wake up, cook breakfast, and head to work; artists paint, while

authors write; they form opinions, notice each other, and initiate conversations; they remember and reflect on days past as they plan the next day. To enable generative agents, we describe an architecture that extends a large language model to store a complete record of the agent's experiences using natural language, synthesizes those memories over time into higher-level reflections, and retrieves them dynamically to plan behavior. We instantiate generative agents to populate an interactive sandbox environment inspired by *The Sims*, where end users can interact with a small town of twenty-five agents using natural language. In an evaluation, these generative agents produce believable individual and emergent social behaviors. For example, starting with only a single user-specified notion that one agent wants to throw a Valentine's Day party, the agents autonomously spread invitations to the party over the next two

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner(s).  
USE27 '23, October 29–November 1, 2023, San Francisco, CA, USA  
© 2023 Copyright held by the owner(s).  
ACM ISBN 978-0-486-14132-9/23/00.  
<https://doi.org/10.1145/3558183.3606763>



**htmaaland.web.app**

# Limitations / Ethics



**“Salmon swimming in the river”**

The New York Times

## An A.I.-Generated Picture Won an Art Prize. Artists Aren't Happy.

"I won, and I didn't break any rules," the artwork's creator says.

The Atlantic

## I Went Viral in the Bad Way

A few lessons from my mistake

By Charlie Warzel

FEATURE

## AI Creating 'Art' Is An Ethical And Copyright Nightmare

If a machine makes art, is it even art? And

Karla Ortiz @kortizart · Aug 13

Curious. Are there any living artists there? Were they added with their consent? Also what about artist's estates? In pretty sure Giger's, Mead's, Kinkadee's, Miura's estates would wish to have a say in this.

Also I KNOW disco diffusion features many of my peers...

 Bokar N'Diaye @bokar\_n · Aug 13

Top 20 dead artists used in @StableDiffusion Discord bot queries as of today, 1-2 PM, among the 773 artists in the collective archive of @sureallabs, @proximasan, @KyrickYoung and @EErratica. Can't wait for the Disco Diffusion 4000+ artists to be added !

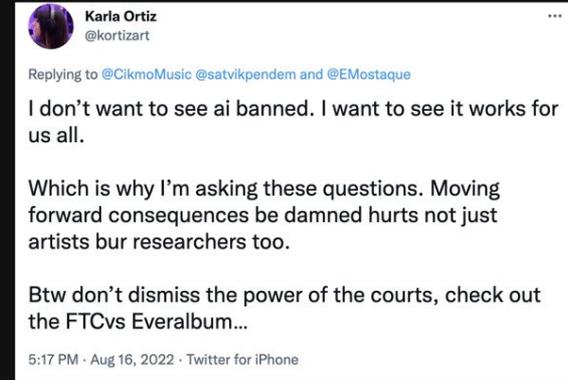
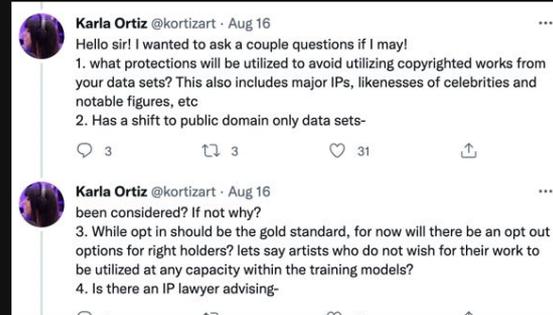
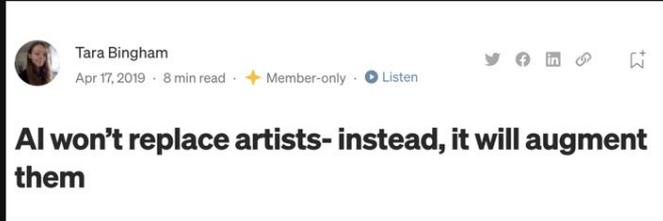
[Show this thread](#)

Artist	Date of Death	Hits on Discord (family name mostly)
Mucha, Alphonse	1939	142674
Beksinski, Zdzislaw	2005	37195
Moebius	2012	28150



MOTHERBOARD  
TECHBYVICE

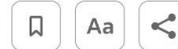
## An AI-Generated Artwork Won First Place at a State Fair Fine Arts Competition, and Artists Are Pissed



# AI-generated art cannot receive copyrights, US court says

By **Blake Brittain**

August 21, 2023 2:29 PM EDT · Updated a month ago



An AI (Artificial Intelligence) sign is seen at the World Artificial Intelligence Conference (WAIC) in Shanghai, China July 6, 2023. REUTERS/Aly Song/File Photo/File Photo [Acquire Licensing Rights](#)

Aug 21 (Reuters) - A work of art created by artificial intelligence without any human input cannot be copyrighted under U.S. law, a U.S. court in Washington, D.C., has ruled.

Only works with human authors can receive copyrights, U.S. District Judge Beryl Howell [said](#) on Friday, affirming the Copyright Office's rejection of an application filed by computer scientist Stephen Thaler on behalf of his DABUS system.

The Friday decision follows [losses](#) for Thaler on bids for U.S. patents covering inventions he said were created by DABUS, short for Device for the Autonomous Bootstrapping of Unified Sentience.

**LLMs make it *easier* to write code / produce artifacts,  
but it is up to humans to *exhibit creativity and  
judgment.***

**Amira  
Abdel-Rahman**

amiraa@mit.edu

**Valdemar  
Danry**

vdanry@mit.edu

**Olivia  
Seow**

oliviaseow@g.harvard.edu