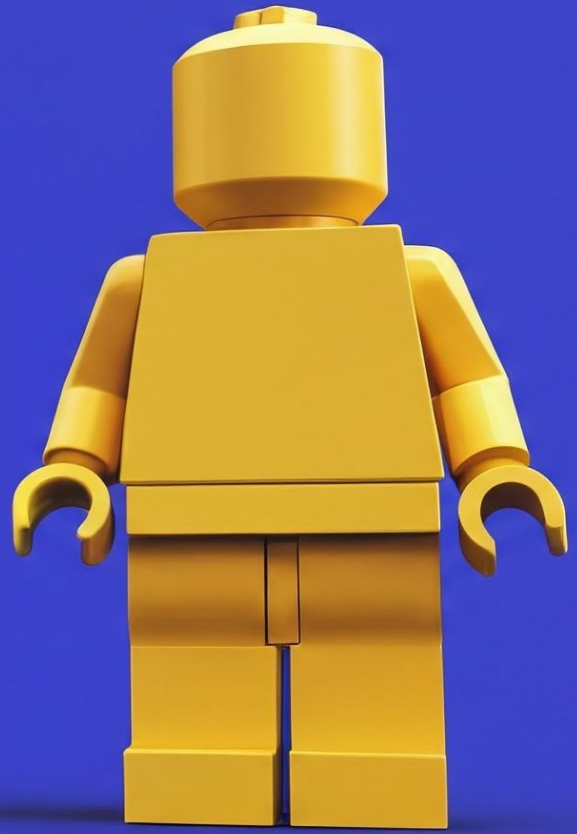
AI.Engineer | World's Fair

UX Design Principles for Semi-Autonomous Multi-Agent Systems



Victor Dibia, PhD | @vykthur
June 5, 2025

Victor Dibia

x.com | linkedIn | newsletter

victordibia.com

Principal RSDE, Microsoft Research

Focused on Human AI Experiences and Agents

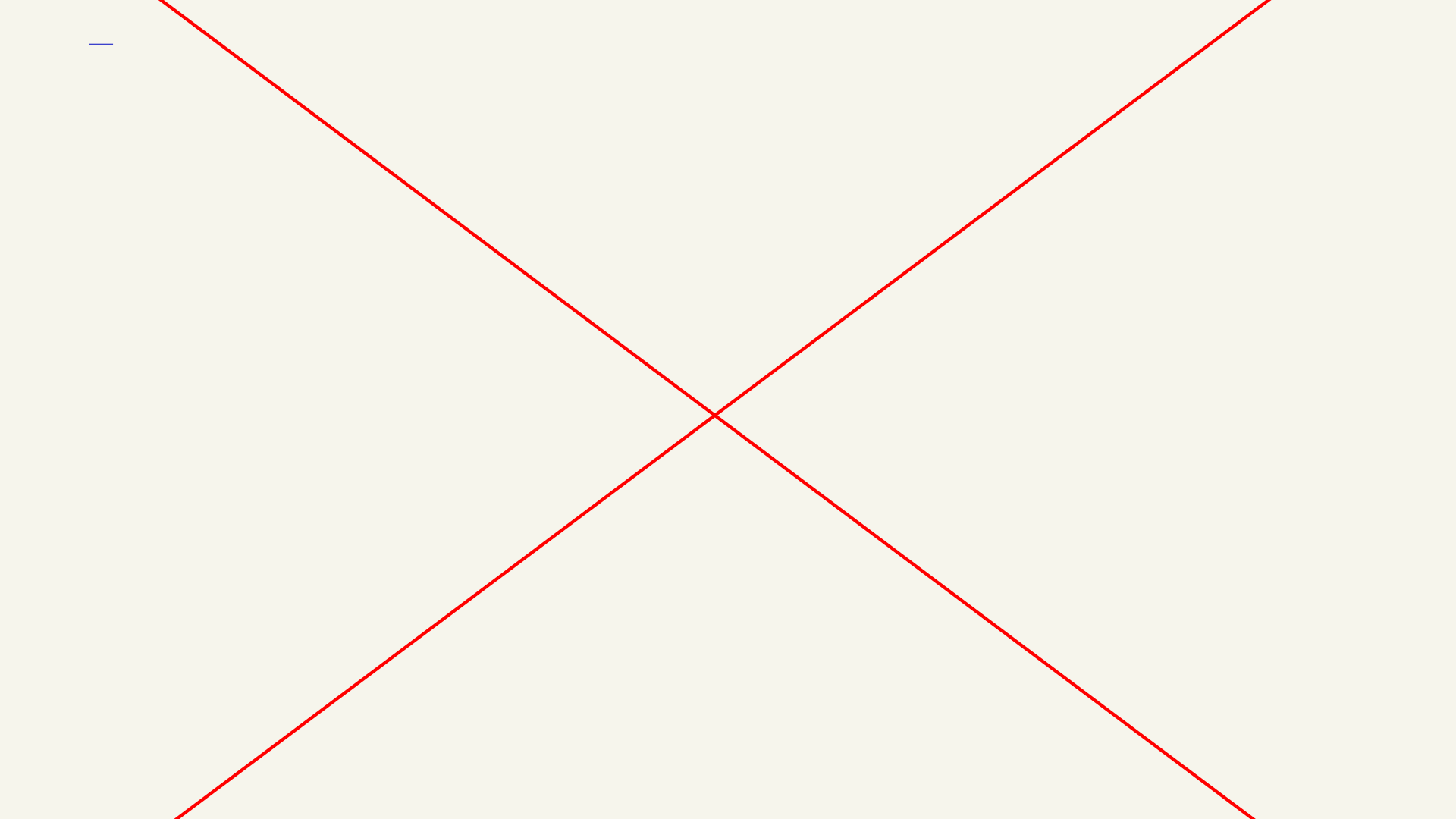
Contributor to GitHub Copilot, AutoGen, AutoGen Studio

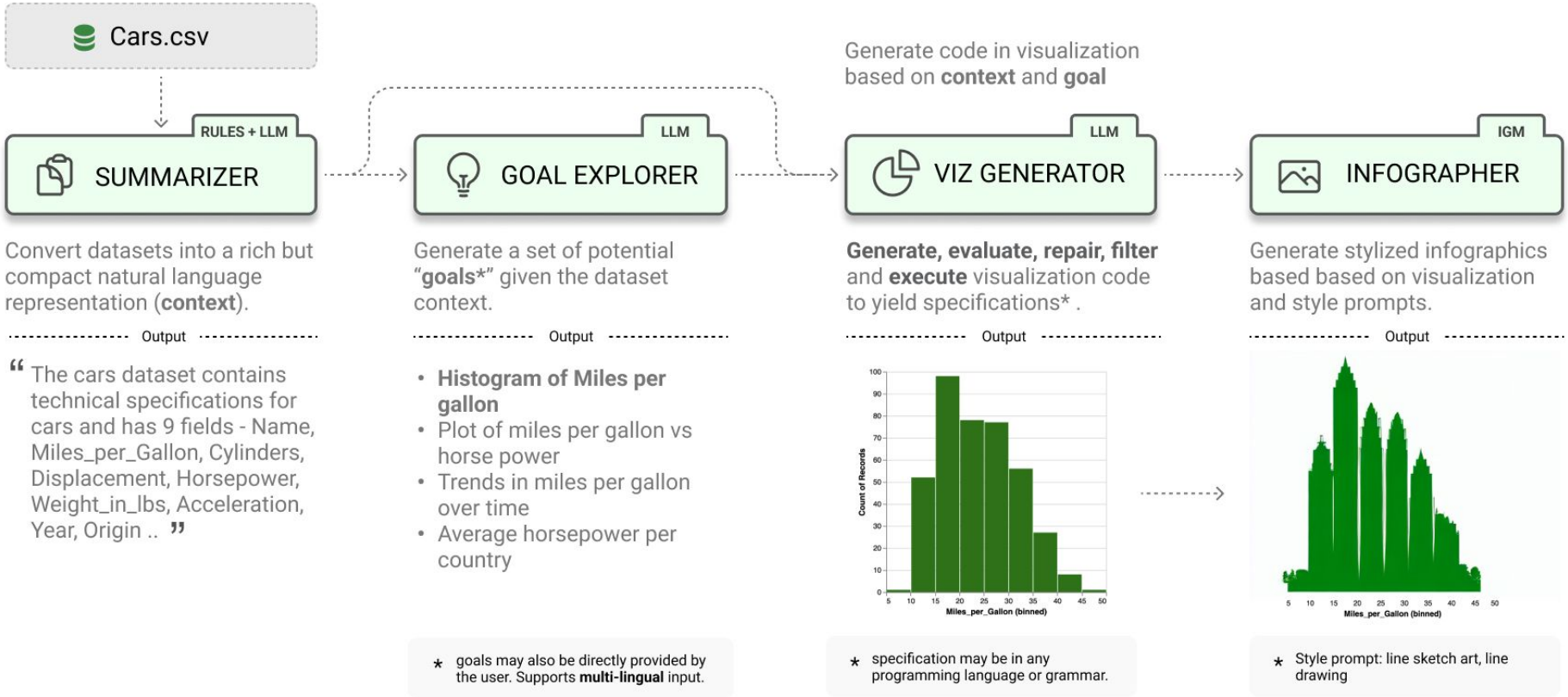
AutoGen - leading OSS framework for building multi-agent applications.

Previously Worked @

- **Cloudera** - ML Engineer
- IBM Research - Research Staff Member

How I got into Agents ..





August 2022 | LIDA: Automatic Generation of Visualizations . https://microsoft.github.io/lida/

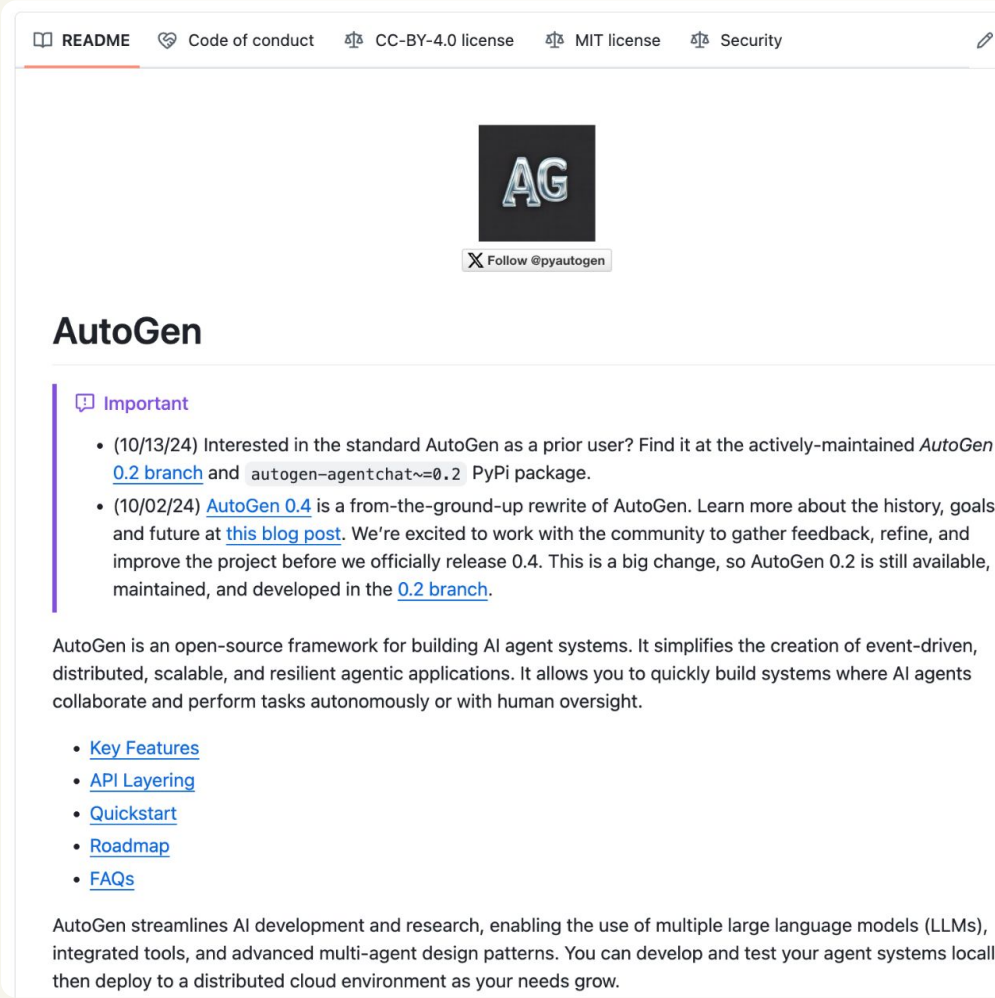
AutoGen

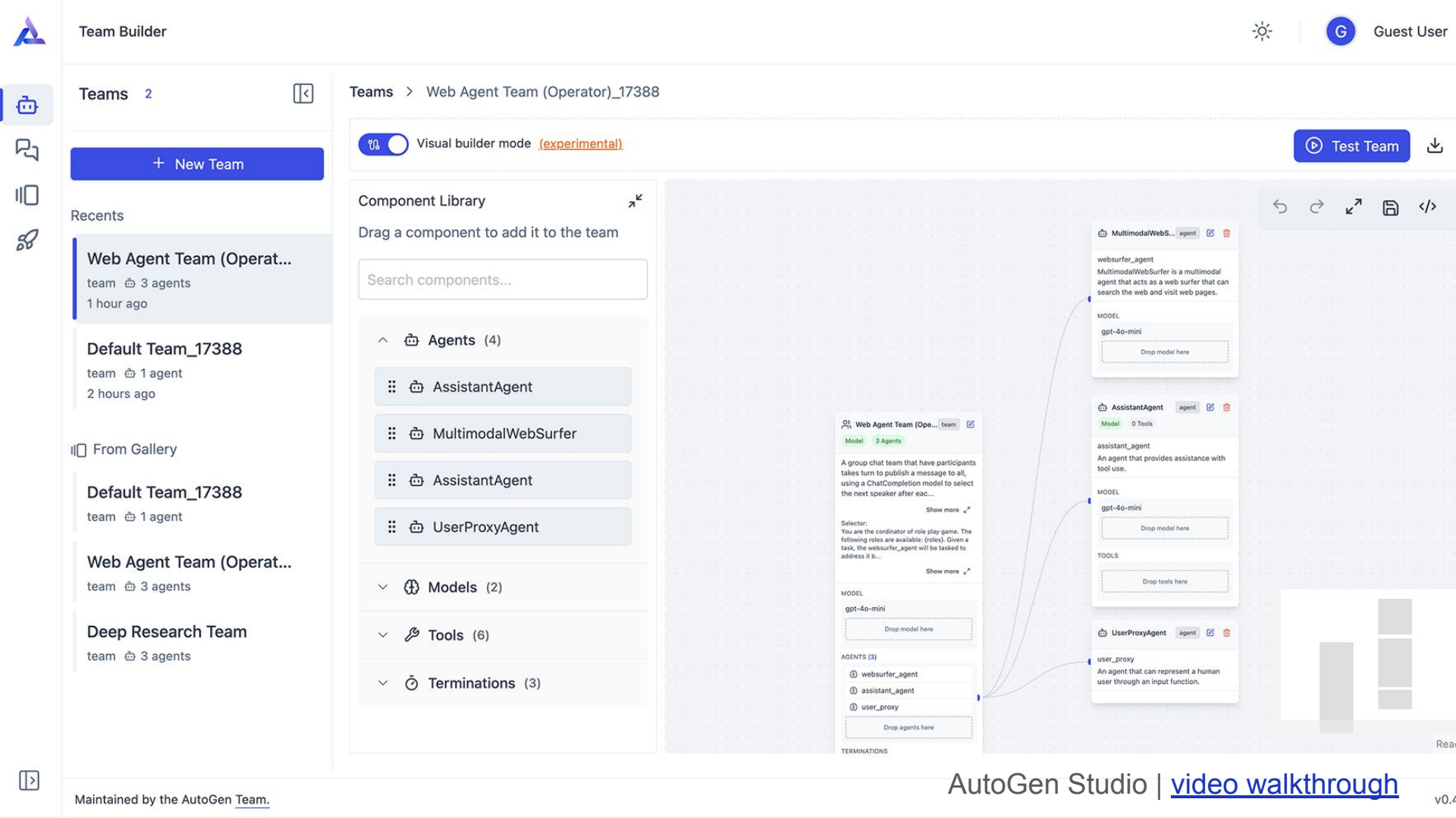
https://github.com/microsoft/autogen

45k stars

An open-source (MIT) framework for building agentic AI applications. Supported in **Azure AI Foundry**

Sept 2023




Team Builder
Guest User
Teams 2
New Team
Recents
Web Agent Team (Operat...
team 3 agents
1 hour ago
Default Team_17388
team 1 agent
2 hours ago
From Gallery
Default Team_17388
team 1 agent
Web Agent Team (Operat...
team 3 agents
Deep Research Team
team 3 agents
Teams > Web Agent Team (Operator)_17388
Visual builder mode (experimental)
Test Team
Component Library
Drag a component to add it to the team
Search components...
Agents (4)
AssistantAgent
MultimodalWebSurfer
AssistantAgent
UserProxyAgent
Models (2)
Tools (6)
Terminations (3)
Web Agent Team (Ope... team
Model 3 Agents
A group chat team that have participants takes turn to publish a message to all, using a ChatCompletion model to select the next speaker after eac...
Show more
Selector:
You are the cordinator of role play game. The following roles are available: {roles}. Given a task, the websurfer_agent will be tasked to address it b...
Show more
MODEL
gpt-4o-mini
Drop model here
AGENTS (3)
websurfer_agent
assistant_agent
user_proxy
Drop agents here
TERMINATIONS
MultimodalWebS... agent
websurfer_agent
MultimodalWebSurfer is a multimodal agent that acts as a web surfer that can search the web and visit web pages.
MODEL
gpt-4o-mini
Drop model here
AssistantAgent agent
Model 0 Tools
assistant_agent
An agent that provides assistance with tool use.
MODEL
gpt-4o-mini
Drop model here
TOOLS
Drop tools here
UserProxyAgent agent
user_proxy
An agent that can represent a human user through an input function.
Maintained by the AutoGen Team.
AutoGen Studio | video walkthrough


AutoGen Studio v0.4
https://github.com/microsoft/autogen

But .. this is ai.engineer

we are all here to learn how to build!

Let’s build something from scratch!

A multi-agent system from scratch!

BlenderLM - a multi-agent system for 3D tasks

https://github.com/victordibia/blenderlm

Blender LM

Connected to Blender

Che

Blender Controls

Chat | Presets | Projects

Execution Plan 5.21s 1936 tokens plan_generated

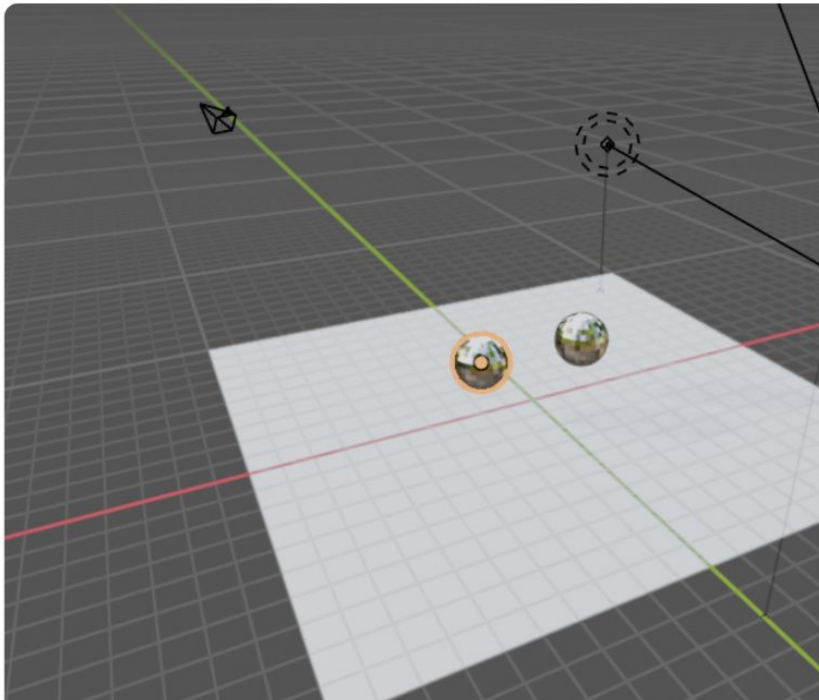
1. Set up the scene environment with a ground plane, complete 3-point lighting (adding fill and rim lights), and a properly positioned camera for clear composition.
2. Create two spheres with correct spatial separation, assign a glossy silver material to both, and position them above the ground plane.

Event: step_start 5.22s

🎯 Step 1/2: Set up the scene environment with a ground plane, complete 3-point lighting (adding fill and rim lights), and a properly positioned camera for

Blender Viewport

Refresh

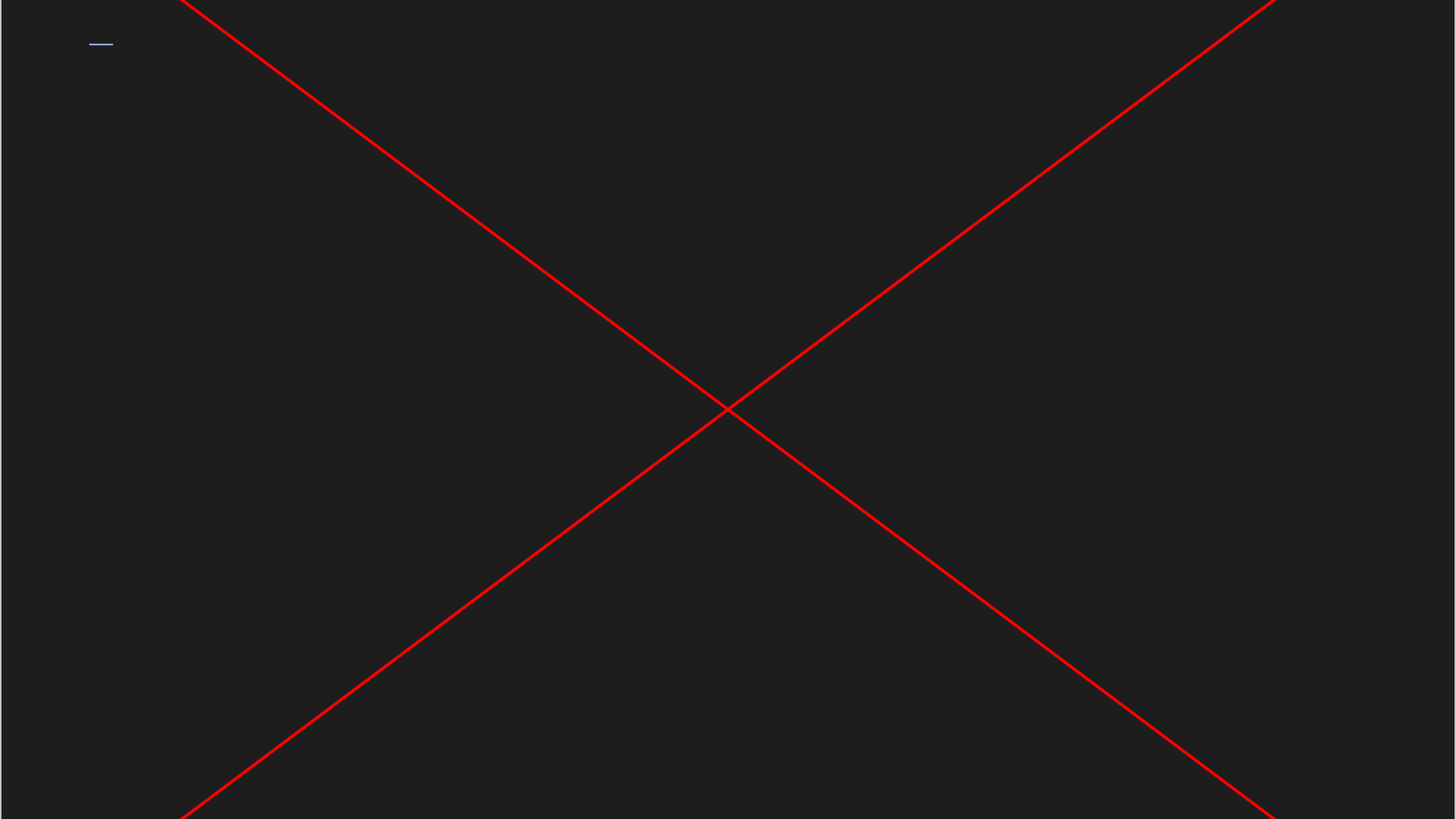


What is the plan for today?

- **Part 0**: I'll show you a **demo**
- **Part 1**: I'll walk you through **how it was built from scratch**
- **Part 2**: We will discuss a set of **design principles** that you can apply to your own agentic applications.
- **Part End**: Takeaways

Demo

Part 0



“Create a scene that contains a low poly well”

How do you build an agentic system for this? ...

Workflow?
- a set of deterministic steps *with sprinkles of LLM calls*

Highly reliable … but they require that we know the exact solution to the task

Workflows help us build reliable, production-ready software

.. but ..

they require that we know the exact solution to the task

Autonomous Multi-Agent Systems?

- an LLM drives control flow of the application

Enables software that takes actions, observes results and interactively explores the solution space

Multi-Agent System

- **Autonomy** | Can do many *different* things
- **Action** | Can take action with *side effects*
- **Duration** | Complex long-running tasks

https://multiagentbook.com

Demo

https://github.com/victordibia/blenderlm

How is it built? Aka - the process

Part 1

Building a Multi-Agent System - *from scratch*

- Goal definition
- Baseline
- Tools
- Eval testbed
- Agent

This comes last, not first!

1. Goal definition

- A system that can **translate natural language tasks to actions to 3D artifacts in Blender**

2. Baseline

- Blender “Hello World” script to “Create a Cube”
  - Blender Addon
  - Client library with Socket connection
- Valuable to test *feasibility* and mechanics of this entire process

3. Tools

Your agent is only as good as the tools you give it!

- Build a set of **task specific** and **general purpose tools** related to the system goals
  - Create object ..
  - Execute code!

```
from blenderlm.client.tools import
create_blender_object, execute_code

result: Any = await create_blender_object
(type="CUBE", name="MyCube", location_x=0,
location_y=0, location_z=0, session_id=None,
wait_for_result=True)

result: Any = await execute_code(code="bpy.data.
objects['MyCube'].location.x += 1",
session_id=None, wait_for_result=True)
```

4. Eval Test Bed

- The ability to run rapid experiments that inform system update decisions
  - **v1** : Jupyter notebook
  - **v2** : a full interactive React web UI with viewport capture / streaming.
  - **v3**: automated test suites, metrics and evaluation harness

Blender LM

Guest User

Connected to Blender

Check Connection

Blender Controls

Chat (Agent) | Tools | Code | Projects

Clear Scene

Delete all objects from the scene

Clear Scene

Add Sphere

Add a sphere with random position

+ Add Sphere

Add Cube

Add a cube with random position

+ Add Cube

Render Scene

Create a full render of the scene

Render

Add Camera

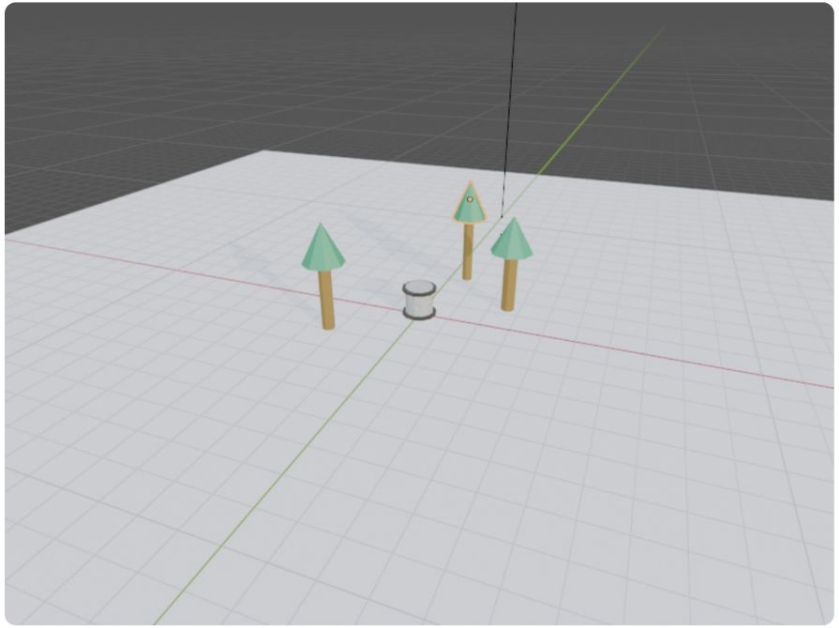
Add a new camera to the scene

+ Add Camera

Job History

Blender Viewport

Refresh | Download



Scene 14 objects

interactive UI to rapidly test capabilities

5. Agents

- Base Agent Loop
  - **LLM** Call (task + prompt)
  - Process **tool** calls in a loop until final results
  - Return Results

```python

from blenderlm.client.agents import OpenAIAgent
from blenderlm.client import get_blender_tools

blender_tool_functions: List[Callable[..., Any]] = await
get_blender_tools()

# Instantiate the agent, passing the tool functions
agent = OpenAIAgent(
    tools=blender_tool_functions,
    model_name="gpt-4.1")

updates: AsyncGenerator[BaseAgentMes... =   agent.run_stream(
    task="add a cube to the scene")
async for update in updates:
    print(update.role,":", update.content, str(update.metadata))

```

5. Agents

- Reliability Improvements
  - **Enriched context** (scene info, blender viewport)
  - **Verifier Agent** (self-eval, retry) | structured output
  - **Planner Agent** (atomic task decomposition) | structured output

Design Principles

These principles are still *early* and non-exhaustive

Part 2

https://multiagentbook.com

Capability Discovery

- Itemize key system capabilities, communicate reliability
- Proactive suggestions based on user context

Help users understand what the agents can do

Try asking:

create a low poly well with two trees

two balls with glossy silver finish

Add a red cube

Create a blue sphere

Add a green cone at [2, 0, 0]

Clear the scene

Observability and Provenance

- Activity log visualizations
- Debugging and provenance tools

Ensure users can observe/trace agent actions

Pro tip: use async generators

Blender LM

Connected to Blender

Che

Blender Controls

Chat | Presets | Projects

Execution Plan 5.21s 1936 tokens plan_generated

1. Set up the scene environment with a ground plane, complete 3-point lighting (adding fill and rim lights), and a properly positioned camera for clear composition.

2. Create two spheres with correct spatial separation, assign a glossy silver material to the ground plane.

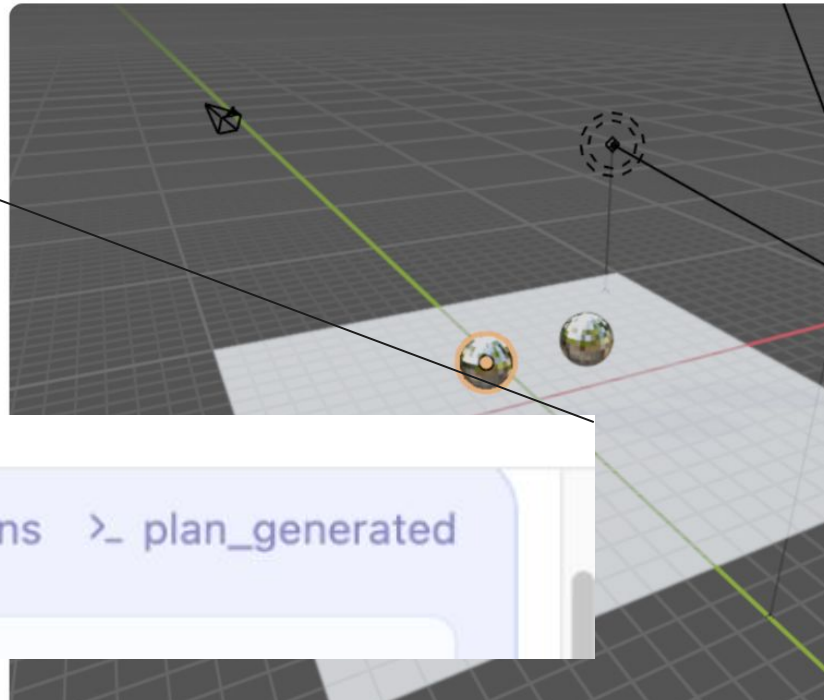
5.21s 1936 tokens plan_generated

Event: step_sta

Step 1/2: Set up ground plane, complete 3-point lighting (adding fill and rim lights), and a properly positioned camera for

Blender Viewport

Refresh



Interruptibility

- Persist agent and application **state**
- Provide **pause**, **resume, cancel** or **feedback** controls

Allow users to pause, resume or cancel agent actions

2 Create two low poly trees with trunk and foliage materials, and position them on either side of the well for balanced composition.

Event: step_start 5.26s

🎯 Step 1/2: Create a low poly well with wooden and stone materials, and position it centrally on the ground plane.

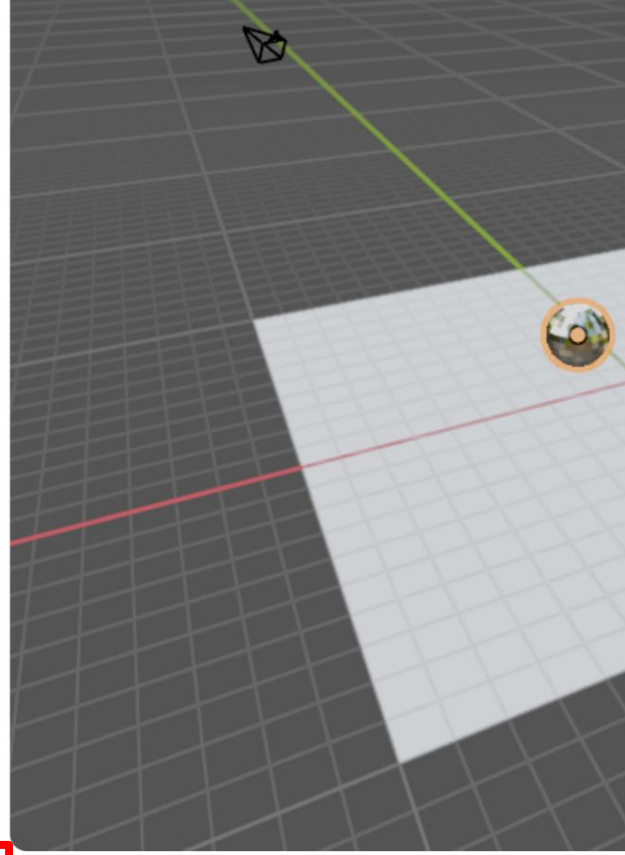
Event: step_execution 5.95s

🔧 Executing step 1/2 (attempt 1)

Blender Assistant

Processing... (step 1/2)

Processing...



Scene 1 objects

Try asking:

Cost-Aware Delegation

- Estimate and communicate the ***cost*** of agent actions
- Provide **controls on when to delegate actions to humans**

Communicate the cost of agent actions, allow users decide when agents can act

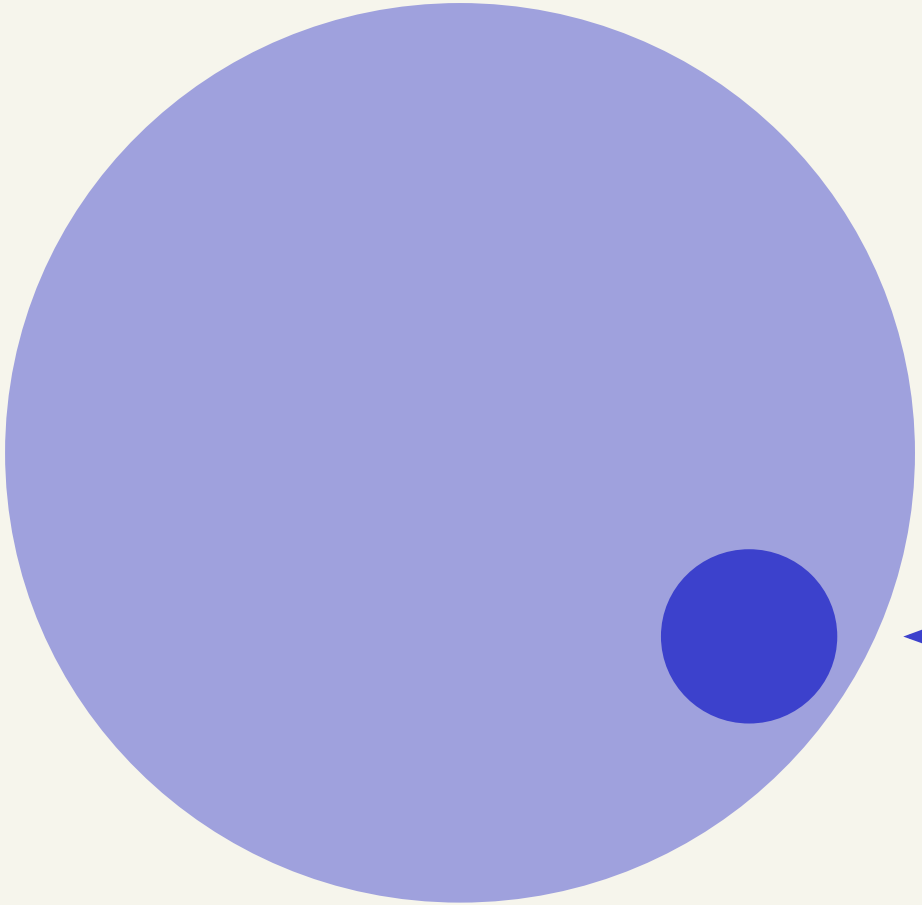
Key Takeaways

Part End

0. Know *when* to use a multi-agent approach!

- Multiple agents collaborating, with autonomy increases the surface for errors and reliability issues
- Like any other tool, they should be selected when they are the **right tool for the job.**

Tasks most teams need to address.



Tasks that benefit from an autonomous multi-agent approach

https://multiagentbook.com

How do I know if my task benefits from a multi-agent approach?

https://multiagentbook.com

A Complex Task Perspective / Checklist

Planning

- Task can be decomposed into a set of steps that lead to a goal state

Adaptive Solution

- Task exists in a dynamic environment, solution is unknown until actions taken

Diverse Perspectives

- Steps in the solution can be mapped into distinct **domains/expertise**

Extensive Context

- Task involves processing extensive context per step

1. Eval driven design

- Define your task
- Define evaluation metrics and test harness
- Build a non-agent baseline
- **Build and improve your agents** and monitor progress on metrics
- Academic benchmarks, while helpful are NOT your task.

2. Human Centered Design

- Ensure users can **discover** the ideal happy path for your agents
- Provide user facing **observability** and traces
- Your agents must be **interruptible**. Ability to pause, resume, give feedback
- Your agents should **quantify the risk/cost** of **actions** and delegate to users as needed

3. Don't build a multi-agent system from scratch for a talk :)

- It's fun, but a lot of work
- If you do, consider a framework (AutoGen) for a quick and dirty prototype!

Further Reading

1. **AutoGen Studio**: A low code tool for building multi-agent applications | Microsoft Research , Arxiv
2. **Magentic-One**: A Generalist Multi-Agent System for Solving Complex Tasks | Microsoft Research , Arxiv
3. **Magentic-UI**, an experimental human-centered web agent | Microsoft Research
4. **Challenges in Human-Agent Communication**. Microsoft Research | Arxiv

Thank You!

- Follow along book (Chapter 3) - **multiagentbook.com**
- BlenderLM code **https://github.com/victordibia/blenderlm**