

666ghj / MiroFish Public

<> Code Issues 122 Pull requests 133 Discussions Actions Projects Security and quality Insights

main 2 Branches 3 Tags Go to file Go to file <> Code

666ghj docs: rename README-EN.md to README.md as default English documentation	fa0f651 · 3 weeks ago	
.github/workflows	feat(docker): add Docker support with Dock...	3 months ago
backend	Merge pull request #428 from Ghostubborn...	last month
frontend	fix(security): upgrade axios, rollup, picomat...	last month
locales	fix(i18n): further shorten English metrics an...	last month
static/image	feat(README): add demo video sections f...	3 months ago
.dockerignore	feat(docker): add Docker support with Dock...	3 months ago
.env.example	refactor(.env.example): update LLM API co...	3 months ago
.gitignore	fix(report_agent): refine tool call handling a...	2 months ago
Dockerfile	feat(docker): add Docker support with Dock...	3 months ago
LICENSE	Add GNU AGPL v3 License	4 months ago
README-ZH.md	docs: rename README.md to README-Z...	3 weeks ago
README.md	docs: rename README-EN.md to READM...	3 weeks ago
docker-compose.yml	feat(docker): add Docker support with Dock...	3 months ago
package-lock.json	Change project license from Apache-2.0 to ...	4 months ago
package.json	Fix: Change backend dev color to green	4 months ago

README AGPL-3.0 license



简洁通用的群体智能引擎，预测万物  
*A Simple and Universal Swarm Intelligence Engine, Predicting Anything*



stars 58k watchers 380 forks 8.9k Docker Build Ask DeepWiki

Discord Join X Follow Instagram Follow

# Overview

MiroFish is a next-generation AI prediction engine powered by multi-agent technology. By extracting seed information from the real world (such as breaking news, policy drafts, or financial signals), it automatically constructs a high-fidelity parallel digital world. Within this space, thousands of intelligent agents with independent personalities, long-term memory, and behavioral logic freely interact and undergo social evolution. You can inject variables dynamically from a "God's-eye view" to precisely deduce future trajectories — **rehearse the future in a digital sandbox, and win decisions after countless simulations.**

- You only need to: Upload seed materials (data analysis reports or interesting novel stories) and describe your prediction requirements in natural language
- MiroFish will return: A detailed prediction report and a deeply interactive high-fidelity digital world

## Our Vision

MiroFish is dedicated to creating a swarm intelligence mirror that maps reality. By capturing the collective emergence triggered by individual interactions, we break through the limitations of traditional prediction:

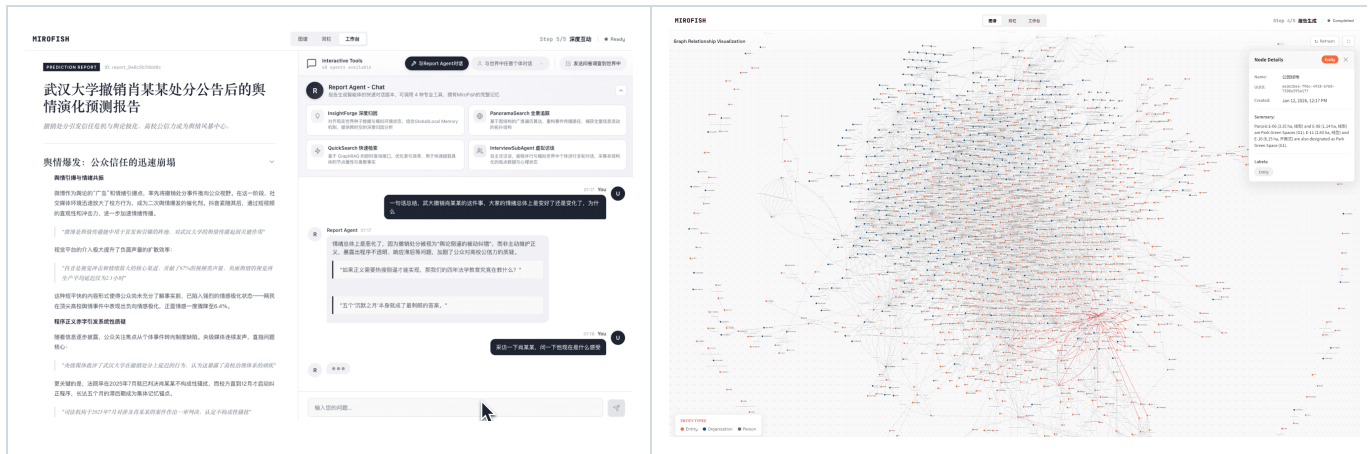
- At the Macro Level:** We are a rehearsal laboratory for decision-makers, allowing policies and public relations to be tested at zero risk
- At the Micro Level:** We are a creative sandbox for individual users — whether deducing novel endings or exploring imaginative scenarios everything can be fun, playful, and accessible

From serious predictions to playful simulations, we let every "what if" see its outcome, making it possible to predict anything.

## Live Demo

Welcome to visit our online demo environment and experience a prediction simulation on trending public opinion events we've prepared for you: [mirofish-live-demo](#)

## Screenshots



## Demo Videos

### 1. Wuhan University Public Opinion Simulation + MiroFish Project Introduction



Click the image to watch the complete demo video for prediction using BettaFish-generated "Wuhan University Public Opinion Report"

### 2. Dream of the Red Chamber Lost Ending Simulation



相当于是时序GraphRAG发挥的作用

Click the image to watch MiroFish's deep prediction of the lost ending based on hundreds of thousands of words from the first 80 chapters of "Dream of the Red Chamber"

Financial Prediction, Political News Prediction and more examples coming soon...

## Workflow

- Graph Building:** Seed extraction & Individual/collective memory injection & GraphRAG construction
- Environment Setup:** Entity relationship extraction & Persona generation & Agent configuration injection
- Simulation:** Dual-platform parallel simulation & Auto-parse prediction requirements & Dynamic temporal memory updates
- Report Generation:** ReportAgent with rich toolset for deep interaction with post-simulation environment
- Deep Interaction:** Chat with any agent in the simulated world & Interact with ReportAgent

## Quick Start

### Option 1: Source Code Deployment (Recommended)

#### Prerequisites

Tool	Version	Description	Check Installation
Node.js	18+	Frontend runtime, includes npm	<code>node -v</code>
Python	≥3.11, ≤3.12	Backend runtime	<code>python --version</code>
uv	Latest	Python package manager	<code>uv --version</code>

#### 1. Configure Environment Variables

```
# Copy the example configuration file
cp .env.example .env

# Edit the .env file and fill in the required API keys
```

#### Required Environment Variables:

```
# LLM API Configuration (supports any LLM API with OpenAI SDK format)
# Recommended: Alibaba Qwen-plus model via Bailian Platform: https://bailian.console.aliyun.com/
# High consumption, try simulations with fewer than 40 rounds first
LLM_API_KEY=your_api_key
LLM_BASE_URL=https://dashscope.aliyuncs.com/compatible-mode/v1
LLM_MODEL_NAME=qwen-plus
```

```
# Zep Cloud Configuration
# Free monthly quota is sufficient for simple usage: https://app.getzep.com/
ZEP_API_KEY=your_zep_api_key
```

## 2. Install Dependencies

```
# One-click installation of all dependencies (root + frontend + backend)
npm run setup:all
```

Or install step by step:

```
# Install Node dependencies (root + frontend)
npm run setup

# Install Python dependencies (backend, auto-creates virtual environment)
npm run setup:backend
```

## 3. Start Services

```
# Start both frontend and backend (run from project root)
npm run dev
```

### Service URLs:

- Frontend: `http://localhost:3000`
- Backend API: `http://localhost:5001`

### Start Individually:

```
npm run backend # Start backend only
npm run frontend # Start frontend only
```

## Option 2: Docker Deployment

```
# 1. Configure environment variables (same as source deployment)
cp .env.example .env

# 2. Pull image and start
docker compose up -d
```

Reads `.env` from root directory by default, maps ports `3000` (frontend) / `5001` (backend)

Mirror address for faster pulling is provided as comments in `docker-compose.yml`, replace if needed.

## Join the Conversation



 **VibeCoding交流群**  
群号: 1043134984  
在这里每个人都是超级个体



The MiroFish team is recruiting full-time/internship positions. If you're interested in multi-agent simulation and LLM applications, feel free to send your resume to: [mirofish@shanda.com](mailto:mirofish@shanda.com)

## Acknowledgments

MiroFish has received strategic support and incubation from Shanda Group!

MiroFish's simulation engine is powered by [OASIS \(Open Agent Social Interaction Simulations\)](#), We sincerely thank the CAMEL-AI team for their open-source contributions!

## Project Statistics

### Star History



### Releases 3

**V0.1.2** Latest  
on Mar 7

[+ 2 releases](#)

### Packages 1

**mirofish**

### Contributors 3

- 666ghj** BaiFu
- Ghostubborn** Hugh Wang
- cursoragent** Cursor Agent

### Languages

