

Bioma: A Multi-Agent Cognition Engine

Bioma is a powerful and flexible framework for building intelligent multi-agent systems. It combines the strengths of behavior trees, utility AI, and large language models (LLMs) to create agents capable of complex decision-making and goal-directed behaviors.

Key Features

- **Behavior Trees:** Bioma utilizes behavior trees as the core structure for defining agent behaviors. Behavior trees provide a modular and hierarchical approach to designing and executing complex agent actions and decision-making processes.
- **Utility AI:** Bioma incorporates utility AI techniques to enable agents to dynamically prioritize and select behaviors based on their current goals, preferences, and environmental factors. This allows for more adaptive and contextually relevant agent behaviors.
- **LLM Integration:** Bioma leverages the power of large language models (LLMs) to enhance agent cognition and decision-making capabilities. LLMs can be used for tasks such as natural language understanding, knowledge retrieval, and generating meaningful responses or actions.
- **Generative and Vision Models:** Bioma integrates generative models and vision models as tools to further expand the capabilities of agents. Generative models can be used for creating new content or solutions, while vision models enable agents to perceive and analyze visual information from their environment.
- **Multi-Agent Support:** Bioma is designed to support multi-agent systems, allowing for the creation and coordination of multiple intelligent agents within a single environment. Agents can communicate, collaborate, and interact with each other to achieve common goals or compete against each other.
- **Extensible Architecture:** Bioma provides an extensible architecture that allows developers to easily integrate new models, behaviors, and capabilities into their agents. The modular design of behavior trees and the flexibility of utility AI enable seamless integration of additional components and features.

Getting Started

To get started with Bioma, follow these steps:

1. Clone the Bioma repository: `git clone https://github.com/BiomaAI/bioma.git`
2. Install the necessary dependencies: `cargo install`

3. Explore the examples and documentation to understand the basic concepts and usage of Bioma.
4. Start building your own intelligent agents using the provided APIs and tools.

Examples

Bioma comes with a set of example agents and scenarios to demonstrate its capabilities. Some notable examples include:

- **ChatBot:** A conversational agent that utilizes LLMs to engage in natural language conversations and provide helpful responses.
- **Autonomous Explorer:** An agent that navigates and explores a virtual environment, making decisions based on its perception and goals.
- **Collaborative Planners:** Multiple agents working together to plan and execute complex tasks by combining their individual capabilities and knowledge.

Documentation

For detailed documentation on how to use Bioma, including API references, tutorials, and best practices, please refer to the Bioma Documentation.

Contributing

We welcome contributions from the community to help improve and expand Bioma. If you'd like to contribute, please follow the guidelines outlined in CONTRIBUTING.md.

Testing

```
RUST_LOG=info,bioma_actor::actor=debug cargo test --release -p bioma_actor -- --nocapture t
RUST_LOG=info,bioma_actor::actor=debug cargo test --release -p bioma_behavior -- --nocapture
```

Examples

```
cargo run --release -p bioma_actor --example tictactoe
cargo run --release -p bioma_llm --example chat
cargo run --release -p bioma_llm --example rerank
cargo run --release -p bioma_llm --example embeddings
cargo run --release -p bioma_llm --example indexer -- --root /path/to/custom/root --globs "*"
cargo run --release -p bioma_llm --example retriever -- --query "What is the meaning of life?"
cargo run --release -p bioma_llm --example rag -- --query "What is the meaning of life?" --r
```

```
cargo run --release -p bioma_actor --example object_store
```

If manually launching surrealdb:

```
surreal start --no-banner --allow-all --bind 0.0.0.0:9123 --user root --pass root surrealkv
```

Rerank for OSX (or non docker):

```
python3 -m venv .bioma
source .bioma/bin/activate
pip install torch transformers flask
python assets/scripts/rerank_server.py
```

RAG server example

Agentic RAG Server

Launch the server:

```
cargo run --release -p bioma_llm --example rag_server
```

Reset the engine:

```
curl -X POST http://localhost:5766/reset
```

Upload files:

```
# Upload a single file
curl -X POST http://localhost:5766/upload \
  -F 'file=@./path/to/file.md' \
  -F 'metadata={"path": "dest/path/file.md"};type=application/json'
```

```
# Upload a zip archive
curl -X POST http://localhost:5766/upload \
  -F 'file=@./archive.zip' \
  -F 'metadata={"path": "archive.zip"};type=application/json'
```

Index files:

```
curl -X POST http://localhost:5766/index \
  -H "Content-Type: application/json" \
  -d '{"globs": ["./path/to/files/**/*.rs"], "chunk_capacity": {"start": 500, "end": 2000}]'
```

Retrieve context:

```
curl -X POST http://localhost:5766/retrieve \
  -H "Content-Type: application/json" \
  -d '{
    "type": "Text",
    "text": "..."
  }'
```

```

    "query": "What is Bioma?",
    "threshold": 0.0,
    "limit": 10,
    "source": ".*"
  }'

```

Ask a question:

```

curl -X POST http://localhost:5766/ask \
  -H "Content-Type: application/json" \
  -d '{"query": "What is Bioma?"}'

```

Generate embeddings:

```

curl -X POST http://localhost:5766/api/embed \
  -H "Content-Type: application/json" \
  -d '{
    "model": "nomic-embed-text",
    "input": [
      "Why is the sky blue?",
      "Why is the grass green?"
    ]
  }'

```

Rerank texts:

```

curl -X POST http://localhost:5766/rerank \
  -H "Content-Type: application/json" \
  -d '{
    "query": "What is Deep Learning?",
    "texts": [
      "Deep Learning is learning under water",
      "Deep learning is a branch of machine learning"
    ],
    "raw_scores": false
  }'

```

Chat completion:

```

curl -X POST http://localhost:5766/api/chat \
  -H "Content-Type: application/json" \
  -d '{
    "model": "llama3.2",
    "messages": [
      {
        "role": "user",
        "content": "Why is the sky blue?"
      }
    ]
  }'

```

```
}  
]  
'
```

Ask a question:

```
curl -X POST http://localhost:5766/ask \  
-H "Content-Type: application/json" \  
-d '{"messages": [{"role": "user", "content": "Why is the sky blue?"}]}'
```

Delete indexed sources:

```
curl -X POST http://localhost:5766/delete_source \  
-H "Content-Type: application/json" \  
-d '{"sources": ["path/to/source1", "path/to/source2"]}'
```

Connect to examples DB:

```
surreal sql -e ws://localhost:9123 -u root -p root --namespace dev --database bioma
```

Clean DB:

```
REMOVE DATABASE bioma;
```