Agent-Friendly Visualizations for the Theory of Everything

Theory of Everything Documentation

Introduction

This module provides enhanced 3D and 4D visualizations for the Theory of Everything that can be called programmatically by AI agents. It includes a clean API for generating visualizations and accessing formulas without requiring user interaction.

Overview

The agent-friendly visualizations extend the Theory of Everything codebase with a programmatic interface that allows AI agents to: 1. Generate high-quality 3D and 4D visualizations 2. Access mathematical formulas and their LaTeX representations 3. Customize visualization parameters 4. Save visualizations to disk for later use

Components

The agent-friendly visualization system consists of two main components: 1. **\`toe_api.py\`**: Clean API for interacting with the Theory of Everything, with built-in visualization capabilities 2. **\`agent_example.py\`**: Example script demonstrating how an AI agent can use the API

API Usage

### Basic Usage \`\`\`python from toe_api import ToEAPI api = ToEAPI() visualizations = api.list_visualizations() formulas = api.list_formulas() unified_action = api.get_formula("unified_action") params = {"mass": 2.0, "grid_size": 30} vis_path = api.generate_visualization("4d_spacetime_curvature", params) \`\`\` ### Available Visualizations The API provides the following visualizations: 1. **4D Spacetime Curvature** (\`4d_spacetime_curvature\`): Visualizes 4D spacetime around massive objects using a 3D surface with color representing the time dimension. Parameters: - \`mass\`: Mass of the central object (in solar masses) - \`grid_size\`: Size of the grid for visualization 2. **Quantum Foam in 3D** (\`quantum_foam_3d\`): Provides a detailed 3D voxel representation of quantum fluctuations at the Planck scale. Parameters: - \`grid_size\`: Size of the grid for visualization - \`amplitude\`: Amplitude of quantum fluctuations - \`frequency\`: Frequency of quantum fluctuations 3. **Extra Dimensions** (\`extra_dimensions_3d\`): Visualizes how extra dimensions are compactified at each point in our 3D space using Calabi-Yau manifold representations. Parameters: - \`num_dimensions\`: Total number of dimensions to represent - \`grid_size\`: Size of the grid for visualization 4. **4D Higgs Field** (\`4d_higgs_field\`): Represents the 4D Higgs field with spontaneous symmetry breaking, using color to show the 4th dimension. Parameters: - \`grid_size\`: Size of the grid for visualization 5. **4D Gauge Field** (\`gauge_field_4d\`): Displays non-Abelian gauge fields with a 3D vector

field visualization, using color to represent field strength. Parameters: - `grid_size`: Size of the grid for visualization 6. **All Visualizations** (`all`): Generates all available visualizations. ### Available Formulas The API provides access to the following formulas: 1. **Unified Action** (`unified_action`): The unified action (master equation) 2. **Gravity Action** (`gravity_action`): The gravity action components 3. **Matter Action** (`matter_action`): The matter action components 4. **Gauge Field Action** (`gauge_action`): The gauge field action components 5. **Quantum Corrections** (`quantum_corrections`): The quantum corrections components 6. **Full Master Equation** (`full_master_equation`): The full master equation

Command Line Interface

The API also provides a command line interface for generating visualizations and accessing formulas: ```bash python toe_api.py visualize 4d_spacetime_curvature --params '{"mass": 2.0, "grid_size": 30}' python toe_api.py formula unified_action python toe_api.py list visualizations python toe_api.py list formulas ```

Example Agent Interaction

The `agent_example.py` script demonstrates how an AI agent can interact with the Theory of Everything API: ```bash python agent_example.py ``` This script shows how to: 1. List available visualizations and formulas 2. Get formula information including LaTeX representations 3. Generate visualizations with custom parameters 4. Save visualizations to disk

Testing the API

The `test_api.py` script provides a simple way to test that the API is working correctly: ```bash python test_api.py ``` This script performs basic tests of the API functionality: 1. Importing the API 2. Creating an API instance 3. Listing available visualizations 4. Retrieving a formula 5. Generating a simple visualization

Implementation Notes

The agent-friendly visualization system is designed to handle the common issues that arise when working with the Theory of Everything codebase: 1. **Math Module Conflict**: The system automatically handles the conflict between the project's `math` directory and Python's built-in `math` module. 2. **Visualization Performance**: The system optimizes visualization performance by adjusting parameters based on the complexity of the visualization. 3. **Error Handling**: The system includes robust error handling to ensure that AI agents can recover from errors. 4. **Clean API**: The system provides a clean, well-documented API that is easy for AI agents to use.

Future Enhancements

Planned future enhancements include: 1. **Interactive Visualizations**: Support for generating interactive visualizations that can be embedded in web applications. 2. **Animation Support**: Support for generating animations that show time evolution of physical systems. 3. **3D Model Export**: Support for exporting visualizations as 3D models that can be viewed in VR/AR applications. 4. **Real-time Collaboration**: Support for real-time collaboration between AI agents and human users.