Monster Mash Project Documentation

Overview

Monster Mash is a web-based application that utilizes a genetic algorithm to simulate the evolution of virtual creatures. Users can select attributes for parent creatures and generate offspring with traits influenced by these attributes. The project combines Python for backend processing and HTML/CSS/JavaScript for the frontend user interface.

Key Features

- Creature Simulation: Simulates the evolution of creatures with customizable genetic traits.
- Genetic Algorithm Implementation: Utilizes genetic algorithms for simulating natural selection and genetic variation.
- Interactive Web Interface: Provides a user-friendly platform to interact with the simulation.
- Dynamic Image Generation: Visualizes creatures based on their genetic attributes.

Technical Overview

Python Backend

Genetic Algorithm Components

- Creature Class: Represents a creature with attributes like color, size, limbs, and height. Each creature has a fitness score calculated based on these traits.
- Fitness Calculation: Assesses the suitability of creatures based on environmental factors, using predefined weightings for each trait.
- Genetic Operations:
    - tournament_selection: Selects the fittest creatures for reproduction.
    - crossover: Combines traits of two parent creatures to create offspring.
    - mutate: Introduces random variations in offspring traits.

Image Processing

- Image Manipulation: Converts numerical representations of creature traits into visual form using Python Imaging Library (PIL) and NumPy.

Flask Web Application

- Route Management: Handles HTTP requests, rendering the web pages, and processing form submissions.
- Data Integration: Translates user input into parameters for the genetic algorithm and returns visual results.

HTML/CSS/JavaScript Frontend

- Bootstrap Interface: Leverages Bootstrap for responsive and aesthetically pleasing design.
- Dynamic Dropdowns: Uses JavaScript to update dropdown options based on previous selections, enhancing user interaction.

Installation and Setup

Prerequisites:
- Install Python 3.x.
- Install Flask, PIL, and NumPy using pip.
- Ensure a stable internet connection for loading Bootstrap CDN.

Running the Application:
- Clone or download the project repository from [repository link].
- Navigate to the project directory in the terminal or command prompt.
- Run `python app.py` to start the Flask server.
- Open a web browser and visit `http://localhost:5000` to access the application.

Usage Guide

Web Interface

Parent Creature Selection:
- Select attributes for Parent 1 and Parent 2 from dropdown menus.
- Attributes include tribe (color), size, limbs, and height.

Generating Offspring:

- Click the "Generate Offspring" button to create a new creature based on the selected attributes.

Viewing Results:

- The application displays images of the parent creatures and the generated offspring, along with their respective attributes.

Behind the Scenes

- The genetic algorithm simulates evolutionary processes by selecting, crossing, and mutating the genetic traits of creatures to produce offspring. The offspring's attributes are a blend of the parents' traits with potential mutations, visualized as images.

API Reference

Creature Class

- __init__(self, image_array, color_code, size, limbs, height): Initializes a new creature instance.
- calculate_fitness(self, environment_factors=None, time_of_day='daytime'): Calculates the fitness score of the creature.

Genetic Algorithm Functions

- tournament_selection(creatures, tournament_size=3): Performs tournament selection among creatures.
- crossover(parent1, parent2, blend_height=10): Generates offspring from two parents.
- mutate(creature, mutation_rate=0.01): Applies random mutations to a creature.

Contributing

- Contributions are welcome! Please fork the repository, make your changes, and submit a pull request.
- For major changes, please open an issue first to discuss what you would like to change.
- Please ensure to update tests as appropriate.

MIT License

Contact

- For any inquiries or contributions, please contact Chijindu Okafor at chijindu12@gmail.com.