SWEN90004 - Daisyworld: Proposal

Group 18

Junrong Wu (1310531), Xintong Liu (1328108), Xiaotian Li (1141181)

1. Overview

Daisyworld is a model that aims to illustrate Gaia hypothesis by demonstrating the interactions between the biota and global environment in an isolated system.

The project replicates the existing NetLogo Daisyworld model in Java 19, and 3 key entities are in the system: Daisyworld as the observer, Daisy as the turtle and Ground as the patch. The replicated model can be compiled with OpenJDK 19. Runtime variables can be configured in `*params.properties*` file, and execute results in `.csv` format can be found in `*data*` directory

2. Design of the existing model

2.1 World Procedure

At start, environment variables including daisy max age, global temperature and solar luminosity are set up. Then daisies are spread randomly in ground patches with a random age below the max age. After spreading the daisy, global temperature is updated. Then, in each tick, each daisy and patch execute their own life cycle, and *global-temperature* is updated.

2.2 Daisy Procedure

At start, a new-born daisy will be 0 years old, and the colour attribute is allocated.Then, in each tick, a daisy can be living or dead. If its age has reached the *max-age*, it dies; otherwise, if the temperature is suitable, it has a probability of growing a daisy of the same colour in a neighbouring patch. And the model checks attached daisy survivability at each tick.

2.3 Ground Patch Procedure

At start, a new-born daisy may be allocated on one patch randomly based on proportion settings. Then, in each time step, a ground patch updates its temperature since heat is absorbed by the daisy located on this patch and will diffuse to neighbours.

3. Design of replicated model

The UML diagram of model design as is shown in *Figure 1.*

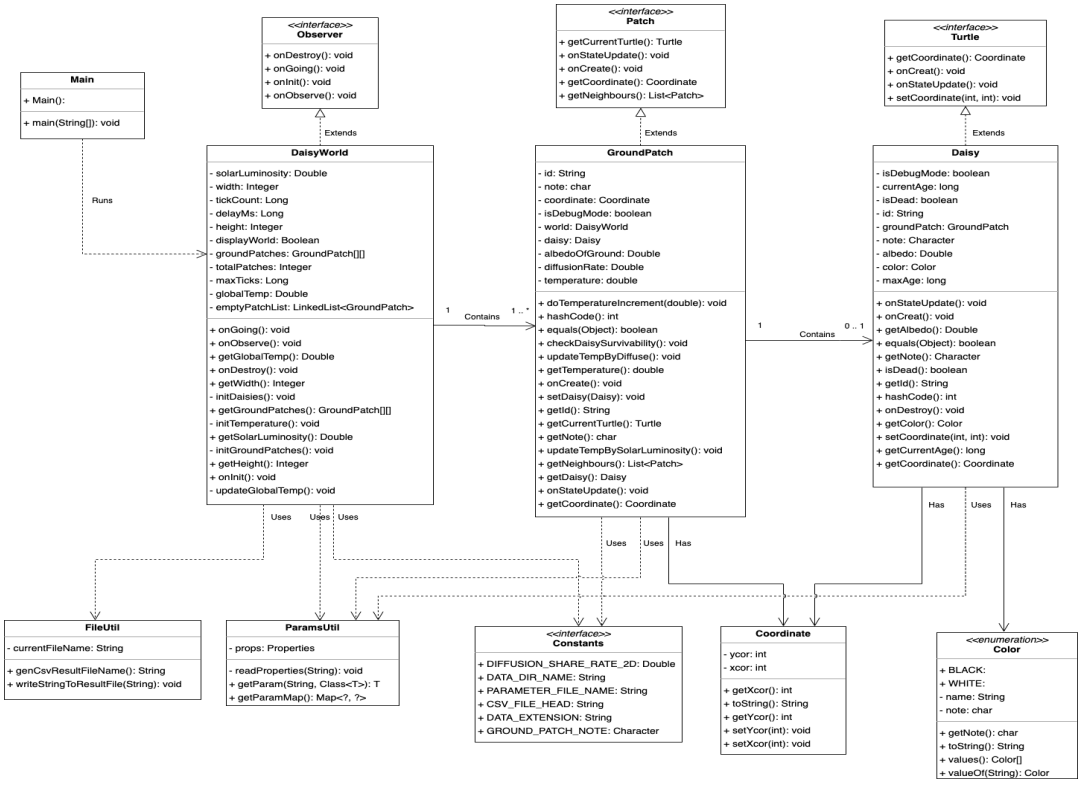

Figure 1. Class Diagram

3.1 Main:

The entrance of the program. **DaisyWorld** instance instantiates and runs here.

3.2 DaisyWorld:

This class represents the Daisyworld in the Netlogo model, which cooperated with the **GroundPatch** and **Daisy** classes. It includes initialization logic (i.e. initDaisies, initTemperature, initGroundPatches) to set up global variables. Meanwhile, some life cycle methods (i.e. onCreate, onGoing, onDestroy, onObserve) keep the world runnable and observable. Finally, it contains some update methods to update world states while the running.

This class contains **GroundPatch** to simulate update procedures happening in the Daisyworld ecosystem.

3.3 GroundPatch:

This class describes the patch in the Daisyworld where the Daisies may live. This class would be used by the **DaisyWorld** object to construct the ground panel for the Daisywolrd. And in each individual ground grid patch, it would contain either a Black/White Daisy or remains empty when the simulation is running. This class first includes a constructor with attributes which describes the properties a single ground grid can hold, such as coordinate, albedo and diffusion rate. It also includes updating methods (i.e. **updateTempBySolarLuminosity()** and **updateTempByDiffuse()**) to update states while the simulation is running. It also contains a method **checkDaisySurvivability()** to check whether the Daisy at this grid is still surviving or not. GroundPatch also handles attached **Daisy** instance's lifecycle, such as age increment, reproduction and survivability checks.

3.4 Daisy:

This class represents a daisy and it includes information like colour, albedo, age and survivability. A daisy's life cycle is attached to a **GroundPatch** instance.

3.5 Coordinate, FileUtil and ParamsUtil:

Coordinate class is a helper class representing the grid coordinate. **FileUtil** class encapsulates the logic for CSV file IO. **ParamsUtil** class encapsulates runtime parameter names, the system can obtain parameters from '**params.properties**' file based on the parameter names provided.

3.6 Others:

An isolated configuration file `**params.properties**` is used to modify environment variables without re-compile the Java source codes, which brings convenience to experiments.

4. Experiments

1) Compare the behaviour of the Netlogo model with replicated Java implementation and see whether they are consistent with each other. During this experiment, we will change some of the parameters within the model i.e. albedo, luminosity to see what the model would behave accordingly and as a result whether our implementation can reflect the actual behaviour of the system to the greatest extent.
2) We plan to simulate global warming by setting a high solar luminosity, by comparing the behaviour with and without mutation, we observe if mutation can reduce global warming better and contribute to the ecological balance of daisy populations.
3) Compare the behaviour after introducing mutation/extension. During the experiment, we will be applying the mutation to the Java implementation, running simulations and collecting data on both the extended and unextended models, afterwards, comparing/analysing the differences between them with the data collected.

Extension:

Gene-mutated daisy: During daisy reproduction, gene mutation can occur, and a new type of daisy can be added to the world, whose albedo differs from that of its parent by adding a Gaussian noise. This extension can reveal how gene-mutated daisies affect the results of this model.

5. Plan

1) Implement the Java Model - Xiaotian Li – Deadline: 10th of May
2) Implement the Java Extension - Junrong Wu, Xintong Liu – Deadline: 12th of May
3) Perform the experiments - All group members – Deadline: 20th of May
4) Report Writing - All group members - Deadline: 25th of May
5) Formatting and Organizing Submission - Xintong Liu - Deadline: 26th of May