

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BÀI TẬP LỚN
HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

**Đề tài: Interactive Simulation of Ecosystem
Food Chains**

Nhóm sinh viên thực hiện :

Vũ Minh Trí 20225940

Hồ Đức Tú 20225676

Nguyễn Đức Trường 20225675

Trần Văn Toàn 20225937

Nguyễn Anh Trà 20215151

Lớp : 151965

Giảng viên hướng dẫn: TS. Nguyễn Thị Thu Trang

Hà Nội, tháng 12 năm 2024

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin trân trọng cảm ơn và bày tỏ lòng biết ơn sâu sắc nhất tới cô **Nguyễn Thị Thu Trang** – Giảng viên Trường Công nghệ thông tin & Truyền thông, Trường Đại học Bách Khoa Hà Nội, giáo viên hướng dẫn bài tập lớn đã nhiệt tình giảng dạy, hướng dẫn, chỉ bảo.

Mặc dù đã cố gắng hoàn thiện sản phẩm nhưng không thể tránh khỏi những thiếu hụt về kiến thức. Chúng em mong muốn nhận được những nhận xét thẳng thắn, chi tiết đến từ thầy để tiếp tục hoàn thiện hơn nữa. Cuối cùng, chúng em xin được gửi lời cảm ơn đến cô Nguyễn Thị Thu Trang đã hướng dẫn chúng em trong suốt quá trình hoàn thiện bài tập lớn. Xin trân trọng cảm ơn cô.

Xin chân thành cảm ơn!

Hà Nội, tháng 12 năm 2024

MỤC LỤC

CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN.....	4
1.1. Mô tả yêu cầu bài toán	4
1.2. Yêu cầu cơ bản.....	4
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ SƠ ĐỒ USECASE VÀ SƠ ĐỒ CLASS	6
2.1. Thiết kế sơ đồ class diagram.....	6
2.1.1 Environment.....	6
2.1.2. Organism.....	7
2.1.3. Animal (Kế thừa Organism)	7
2.1.4. Herbivore (Kế thừa Animal)	8
2.1.5. Carnivore (Kế thừa Animal)	8
2.1.6. Plant (Kế thừa Organism)	9
2.2. Thiết kế sơ đồ use case	9
2.2.1. Use Case: Move to New Slot	10
2.2.2. Use Case: Breed New Organism.....	10
2.2.3. Use Case: Eat Food	11
2.2.4. Use Case: Detect Surrounding Organism	11
2.2.5. Use Case: Evade Predators	12
2.2.6. Use Case: Pursuit Preys	12
2.2.7. Use Case: Grow	13
2.2.8. Use Case: Simulate Ecosystem	13
CHƯƠNG 3. DEMO VÀ ĐÁNH GIÁ KẾT QUẢ.....	14
3.1. DEMO.....	14
3.2. Đánh giá kết quả	16

CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN

1.1. Mô tả yêu cầu bài toán

Dự án mô phỏng một hệ sinh thái gồm nhà sản xuất (plants), động vật ăn cỏ (herbivores) và động vật ăn thịt (carnivores). Mục tiêu là giúp người dùng hiểu được cách năng lượng được truyền qua các bậc dinh dưỡng (trophic levels) và tác động của động lực quần thể đến sự ổn định của hệ sinh thái.

Kiến thức cơ bản:

- Cấu trúc của chuỗi thức ăn:
- Nhà sản xuất (Producers): Chuyển hóa năng lượng mặt trời thông qua quang hợp.
- Động vật tiêu thụ cấp 1 (Herbivores): Ăn nhà sản xuất.
- Động vật tiêu thụ cấp 2 (Carnivores): Ăn động vật ăn cỏ.

Truyền năng lượng:

- Chỉ khoảng 10% năng lượng được truyền từ bậc dinh dưỡng này sang bậc tiếp theo (Quy tắc 10%).

Động lực quần thể:

- Mối quan hệ giữa kẻ săn mồi - con mồi ảnh hưởng đến tỷ lệ sinh, tỷ lệ tử và sự ổn định của quần thể.

1.2. Yêu cầu cơ bản

Giao diện người dùng:

Menu chính:

- Chọn các kịch bản mô phỏng khác nhau:
 - **Hệ sinh thái cân bằng:** Tương tác ổn định giữa các loài.
 - **Bùng nổ dân số:** Mất cân bằng nghiêm trọng khi một loài chiếm ưu thế.
 - **Tuyệt chủng:** Một loài bị mất đột ngột và ảnh hưởng đến hệ sinh thái.
 - **Nút Help:** Giải thích các quy tắc mô phỏng và khái niệm sinh học.
 - **Nút Thoát:** Yêu cầu xác nhận trước khi thoát chương trình.

Màn hình mô phỏng:

- Lưới hệ sinh thái: Mỗi ô lưới đại diện cho một vị trí trong hệ sinh thái và có thể chứa một loài sinh vật:

- Nhà sản xuất: Hiện thị dưới dạng biểu tượng nhỏ màu xanh lá. Không di chuyển và có khả năng phát sinh ở ô trống sau khoảng thời gian cố định.
- Động vật ăn cỏ: Biểu tượng động vật cỡ trung bình, di chuyển đến ô có nhà sản xuất gần nhất.
- Động vật ăn thịt: Biểu tượng lớn hơn, di chuyển đến ô có động vật ăn cỏ gần nhất.

Cơ chế di chuyển:

- Động vật di chuyển dựa trên khoảng cách tới nguồn thức ăn.
- Tốc độ di chuyển có thể được điều chỉnh.

Cơ chế sinh sản:

- Sinh sản dựa trên mức năng lượng của sinh vật:
- Khi năng lượng vượt quá ngưỡng, sinh vật mới được tạo ra ở ô trống bên cạnh.

Cơ chế tương tác:

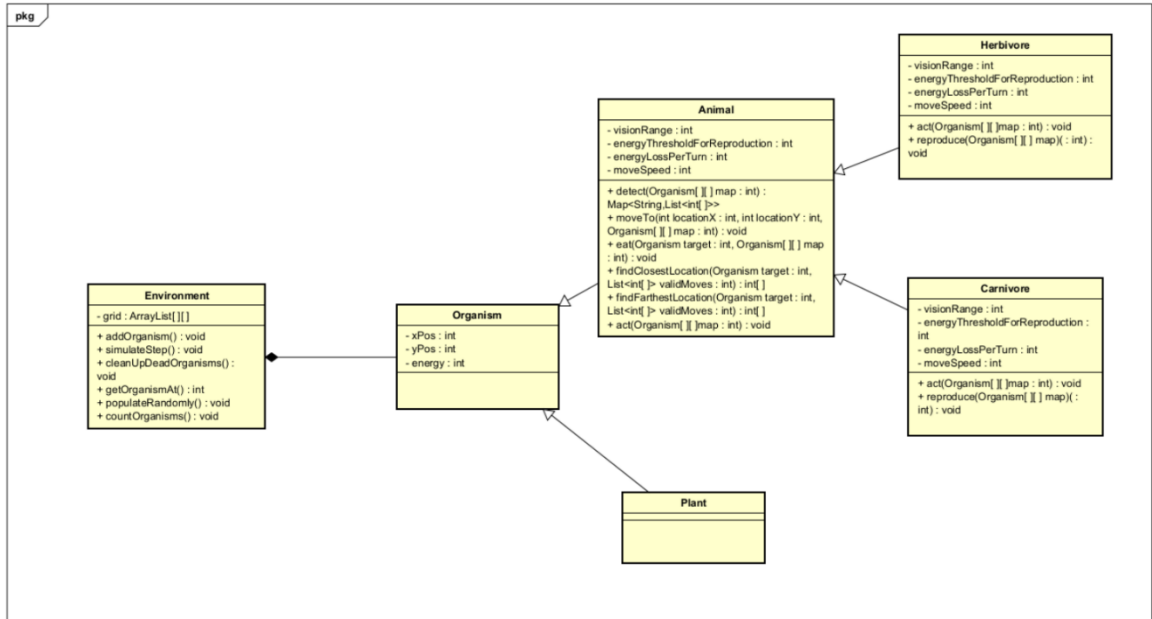
- Nhà sản xuất: Tạo năng lượng thông qua quang hợp.
- Động vật ăn cỏ: Tiêu thụ nhà sản xuất để lấy năng lượng. Nếu không có thức ăn, chúng mất năng lượng dần và có thể chết.
- Động vật ăn thịt: Săn động vật ăn cỏ để lấy năng lượng. Nếu không có con mồi, chúng mất năng lượng và có thể chết.

Truyền năng lượng:

- Nhà sản xuất chuyển hóa ánh sáng mặt trời thành năng lượng.
- Động vật ăn cỏ nhận 10% năng lượng từ nhà sản xuất.
- Động vật ăn thịt nhận 10% năng lượng từ động vật ăn cỏ.

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ SƠ ĐỒ USECASE VÀ SƠ ĐỒ CLASS

2.1. Thiết kế sơ đồ class diagram.



Tổng quan thiết kế:

- Sơ đồ lớp mô tả cấu trúc hệ thống mô phỏng hệ sinh thái với các thành phần chính bao gồm:
 - Environment: Quản lý toàn bộ môi trường và các sinh vật.
 - Organism: Lớp cơ sở chung cho tất cả sinh vật trong hệ sinh thái.
 - Animal (kế thừa Organism): Lớp trung gian chứa các thuộc tính và hành vi của động vật.
 - Herbivore (kế thừa Animal): Động vật ăn cỏ.
 - Carnivore (kế thừa Animal): Động vật ăn thịt.
 - Plant (kế thừa Organism): Thực vật trong môi trường.

Mô tả chi tiết các lớp:

2.1.1 Environment

Chức năng chính: Quản lý môi trường và điều khiển các bước mô phỏng.

Thuộc tính:

- grid: ArrayList[][]
- Dữ liệu dạng lưới (grid) lưu trữ các sinh vật (Organism) tại từng vị trí.

Phương thức:

- addOrganism(): Thêm sinh vật vào lưới.
- simulateStep(): Thực hiện một bước mô phỏng hệ sinh thái, cập nhật hành vi của các sinh vật.
- cleanUpDeadOrganisms(): Xóa các sinh vật đã chết khỏi lưới.
- getOrganismAt(): Lấy thông tin về sinh vật tại một vị trí cụ thể.
- populateRandomly(): Khởi tạo ngẫu nhiên các sinh vật trong lưới.
- countOrganisms(): Đếm số lượng sinh vật còn sống trong môi trường.

2.1.2. Organism

Chức năng chính: Lớp cơ sở chung cho tất cả sinh vật.

Thuộc tính:

- xPos: int - Tọa độ x trên lưới.
- yPos: int - Tọa độ y trên lưới.
- energy: int - Năng lượng hiện tại của sinh vật.

Ghi chú:

Đây là lớp cha cơ sở, các lớp như Animal và Plant sẽ kế thừa từ đây.

2.1.3. Animal (Kế thừa Organism)

Chức năng chính: Lớp trung gian mô tả các thuộc tính và hành vi của động vật.

Thuộc tính:

- visionRange: int - Phạm vi tầm nhìn của động vật.
- energyThresholdForReproduction: int - Ngưỡng năng lượng để động vật có thể sinh sản.
- energyLossPerTurn: int - Lượng năng lượng mất đi mỗi lượt hoạt động.
- moveSpeed: int - Tốc độ di chuyển của động vật.

Phương thức:

- detect(Organism[][] map : int): Phát hiện sinh vật xung quanh trong phạm vi tầm nhìn.
- moveTo(int locationX, int locationY, Organism[][] map : int): Di chuyển đến vị trí cụ thể trong môi trường.

- eat(Organism target, Organism[][] map : int): Tiêu thụ thức ăn (ví dụ: Herbivore ăn Plant, Carnivore ăn Herbivore).
- findClosestLocation(Organism target : int, List<int[]> validMoves : int[]): Tìm vị trí gần nhất đến mục tiêu.
- findFarthestLocation(Organism target : int, List<int[]> validMoves : int[]): Tìm vị trí xa nhất từ mục tiêu.
- act(Organism[][] map : int): Thực hiện hành vi trong một lượt mô phỏng (di chuyển, ăn, sinh sản, v.v.).

Ghi chú

Các lớp Herbivore và Carnivore sẽ kế thừa từ Animal và có thể ghi đè phương thức act() để biểu diễn hành vi cụ thể.

2.1.4. Herbivore (Kế thừa Animal)

Chức năng chính: Mô tả động vật ăn cỏ và hành vi cụ thể của chúng.

Phương thức

- act(Organism[][] map : int):
- Di chuyển đến vị trí có Plant gần nhất.
- Ăn Plant để tăng năng lượng.
- Kiểm tra năng lượng để thực hiện sinh sản nếu đạt ngưỡng.
- Chạy trốn khỏi động vật ăn thịt
- reproduce(Organism[][] map : int): Sinh sản khi đạt đủ năng lượng và có ô trống xung quanh.

2.1.5. Carnivore (Kế thừa Animal)

Chức năng chính: Mô tả động vật ăn thịt và hành vi cụ thể của chúng.

Phương thức:

- act(Organism[][] map : int):
- Di chuyển đến vị trí có Herbivore gần nhất.
- Tiêu thụ Herbivore để tăng năng lượng.
- Kiểm tra năng lượng để thực hiện sinh sản nếu đạt ngưỡng.
- Truy đuổi động vật ăn cỏ.
- reproduce(Organism[][] map : int): Sinh sản khi đạt đủ năng lượng và có ô trống xung quanh.

2.1.6. Plant (Kế thừa Organism)

Chức năng chính: Đại diện cho thực vật trong môi trường.

Ghi chú:

Plant có thể tự động sinh trưởng (tạo Plant mới) tại các ô trống sau một khoảng thời gian nhất định.

Phân tích mối quan hệ giữa các lớp

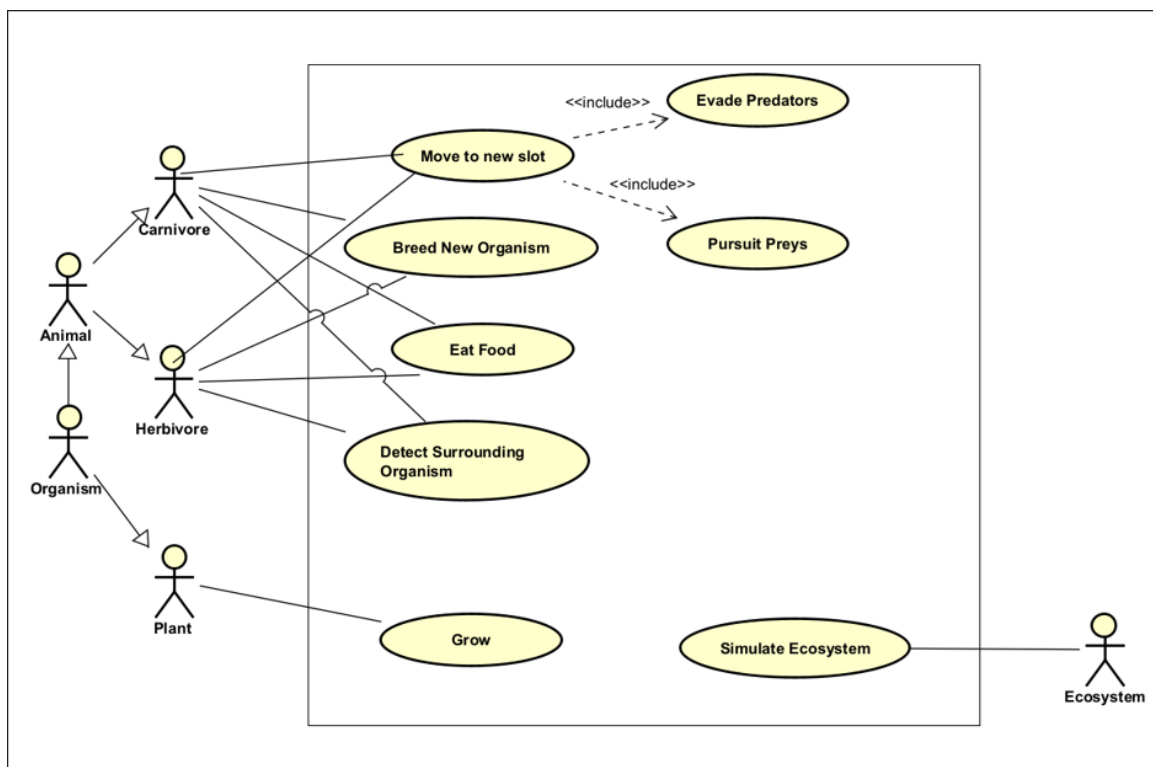
Quan hệ kế thừa:

- Animal kế thừa từ Organism.
- Herbivore và Carnivore kế thừa từ Animal.
- Plant kế thừa từ Organism.

Quan hệ kết hợp:

- Environment sử dụng Organism thông qua thuộc tính grid để quản lý vị trí của sinh vật.
- Các lớp Herbivore và Carnivore tương tác với Plant hoặc các Organism khác (thông qua phương thức eat() và detect()).

2.2. Thiết kế sơ đồ use case



Đặc tả usecase:

2.2.1. Use Case: Move to New Slot

Mô tả:

- Sinh vật (Herbivore và Carnivore) thực hiện di chuyển từ ô hiện tại đến ô mới.
- Quá trình di chuyển có thể là ngẫu nhiên hoặc có chủ đích (ví dụ: Herbivore tìm kiếm thực vật, Carnivore truy đuổi Herbivore).

Actor liên quan:

- Herbivore
- Carnivore

Hoạt động chính:

- Sinh vật xác định vị trí hiện tại và các ô lân cận.
- Kiểm tra điều kiện di chuyển (ô có trống hay không).

Nếu Herbivore:

- Di chuyển về ô có Plant gần nhất.
- Nếu không có Plant, thực hiện di chuyển ngẫu nhiên.

Nếu Carnivore:

- Di chuyển về ô có Herbivore gần nhất.
- Nếu không tìm thấy Herbivore, thực hiện di chuyển ngẫu nhiên.
- Cập nhật trạng thái của sinh vật trên lưới sau khi di chuyển.

2.2.2. Use Case: Breed New Organism

Mô tả:

- Sinh vật tạo ra một sinh vật mới khi đạt ngưỡng năng lượng và có ô trống xung quanh.
- Sinh sản áp dụng cho cả Herbivore và Carnivore.

Actor liên quan:

- Herbivore
- Carnivore

Hoạt động chính:

- Sinh vật kiểm tra năng lượng hiện tại của bản thân.
- Nếu năng lượng vượt quá ngưỡng sinh sản, kiểm tra các ô xung quanh:

- Tìm ô trống để đặt sinh vật mới.
- Tạo một sinh vật mới cùng loại (Herbivore tạo Herbivore, Carnivore tạo Carnivore).
- Giảm năng lượng của sinh vật cha sau khi sinh sản.
- Thêm sinh vật mới vào hệ sinh thái và cập nhật trạng thái.

2.2.3. Use Case: Eat Food

Mô tả:

- Sinh vật tiêu thụ thức ăn để tăng năng lượng.
- Herbivore ăn Plant, Carnivore ăn Herbivore.

Actor liên quan:

- Herbivore
- Carnivore

Hoạt động chính:

Herbivore:

- Xác định ô chứa Plant gần nhất.
- Di chuyển đến ô có Plant và tiêu thụ nó.
- Tăng năng lượng và loại bỏ Plant khỏi ô.

Carnivore:

- Xác định ô chứa Herbivore gần nhất.
- Di chuyển đến ô có Herbivore và tiêu thụ nó.
- Tăng năng lượng và loại bỏ Herbivore khỏi ô.
- Nếu không tìm thấy thức ăn, sinh vật sẽ mất năng lượng theo thời gian.
- Kiểm tra năng lượng sau khi ăn để xác định khả năng sinh sản hoặc sống sót.

2.2.4. Use Case: Detect Surrounding Organism

Mô tả:

- Sinh vật phát hiện các sinh vật khác trong phạm vi xung quanh để thực hiện hành vi phù hợp (di chuyển, ăn, tránh kẻ thù).

Actor liên quan:

- Herbivore

- Carnivore

Hoạt động chính:

- Sinh vật xác định phạm vi phát hiện (ô xung quanh trong một bán kính nhất định).
- Kiểm tra các ô trong phạm vi:
- Herbivore: Tìm kiếm Plant để di chuyển và ăn.
- Carnivore: Tìm kiếm Herbivore để truy đuổi.
- Herbivore cũng phát hiện Carnivore gần đó để thực hiện hành vi tránh né.
- Cập nhật thông tin phát hiện và thực hiện hành động tương ứng.

2.2.5. Use Case: Evade Predators

Mô tả:

- Herbivore phát hiện kẻ săn mồi (Carnivore) trong phạm vi và thực hiện hành vi tránh né để tăng cơ hội sống sót.

Actor liên quan:

- Herbivore

Hoạt động chính:

- Herbivore xác định phạm vi xung quanh và phát hiện sự hiện diện của Carnivore.
- Xác định hướng an toàn nhất để di chuyển (ô không có Carnivore và trống).
- Di chuyển đến ô an toàn.
- Nếu không tìm thấy ô an toàn, Herbivore thực hiện di chuyển ngẫu nhiên.

2.2.6. Use Case: Pursuit Preys

Mô tả:

- Carnivore phát hiện và truy đuổi con mồi (Herbivore) trong phạm vi để tiêu thụ nó và tăng năng lượng.

Actor liên quan:

- Carnivore

Hoạt động chính:

- Carnivore xác định phạm vi xung quanh và tìm Herbivore gần nhất.
- Tính toán đường đi tối ưu để di chuyển đến ô có Herbivore.
- Di chuyển về ô chứa con mồi.
- Tiêu thụ con mồi (Herbivore) khi đến ô đó và tăng năng lượng.
- Loại bỏ con mồi khỏi hệ sinh thái.

2.2.7. Use Case: Grow

Mô tả:

- Plant thực hiện hành vi sinh trưởng bằng cách tạo ra cây mới trong ô trống sau một khoảng thời gian.

Actor liên quan:

- Plant

Hoạt động chính:

- Plant xác định thời gian đã trôi qua từ lần sinh trưởng trước.
- Nếu đủ thời gian, kiểm tra các ô xung quanh để tìm ô trống.
- Sinh trưởng (tạo Plant mới) trong ô trống.
- Reset bộ đếm thời gian sinh trưởng.

2.2.8. Use Case: Simulate Ecosystem

Mô tả:

- Mô phỏng toàn bộ hệ sinh thái qua các vòng lặp, cập nhật trạng thái của từng sinh vật và quản lý các tương tác giữa chúng.

Actor liên quan:

- Ecosystem

Hoạt động chính:

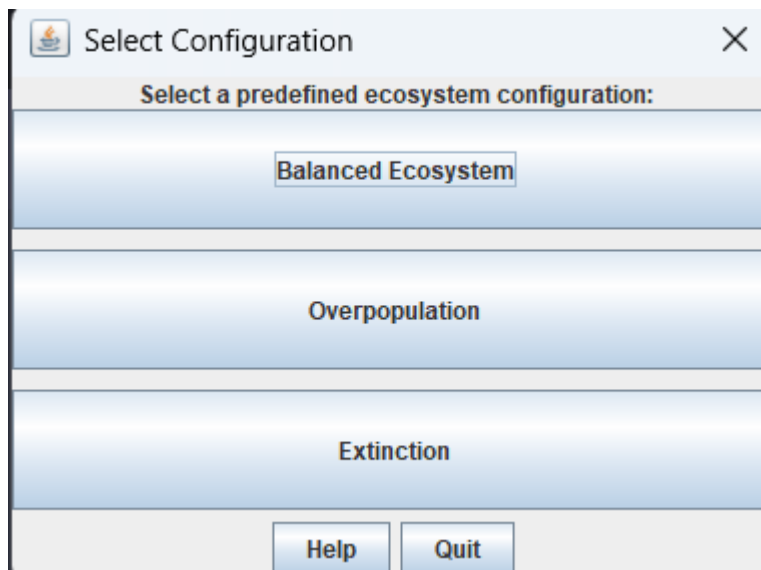
- Hệ thống khởi tạo và đặt các sinh vật vào lưới.
- Trong mỗi vòng lặp mô phỏng:
- Cập nhật trạng thái của Plant (Grow).
- Cập nhật trạng thái của Herbivore (Move, Eat, Evade Predators, Breed).
- Cập nhật trạng thái của Carnivore (Move, Pursuit Preys, Eat, Breed).
- Kiểm tra và loại bỏ sinh vật chết (do hết năng lượng).

- Hiển thị trạng thái hệ sinh thái (số lượng sinh vật, năng lượng trao đổi, v.v.).
- Kết thúc mô phỏng khi đạt điều kiện dừng (ví dụ: tất cả sinh vật chết hoặc hết thời gian).

CHƯƠNG 3. DEMO VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. DEMO

Khi khởi động chương trình người dùng sẽ thấy xuất hiện 1 cửa sổ tại đây người dùng có thể lựa chọn 1 trong các kịch bản mô phỏng hoặc là help hoặc là quit



Sau khi lựa chọn xong thì sẽ chuyển sang cửa sổ dòng lệnh khác tại đây người dùng có thể nhập thay đổi các thông số

Simulation Setup

Grid Width: 30

Grid Height: 30

Plant Spawn Rate: 0.01

initial Plants: 50

initial Herbivores: 30

initial Carnivores: 5

Carnivore Speed: 2

Carnivore Range: 5

Carnivore Energy Threshold: 200

Carnivore Energy Decay: 5

Carnivore Default Energy: 200

Herbivore Speed: 2

Herbivore Range: 5

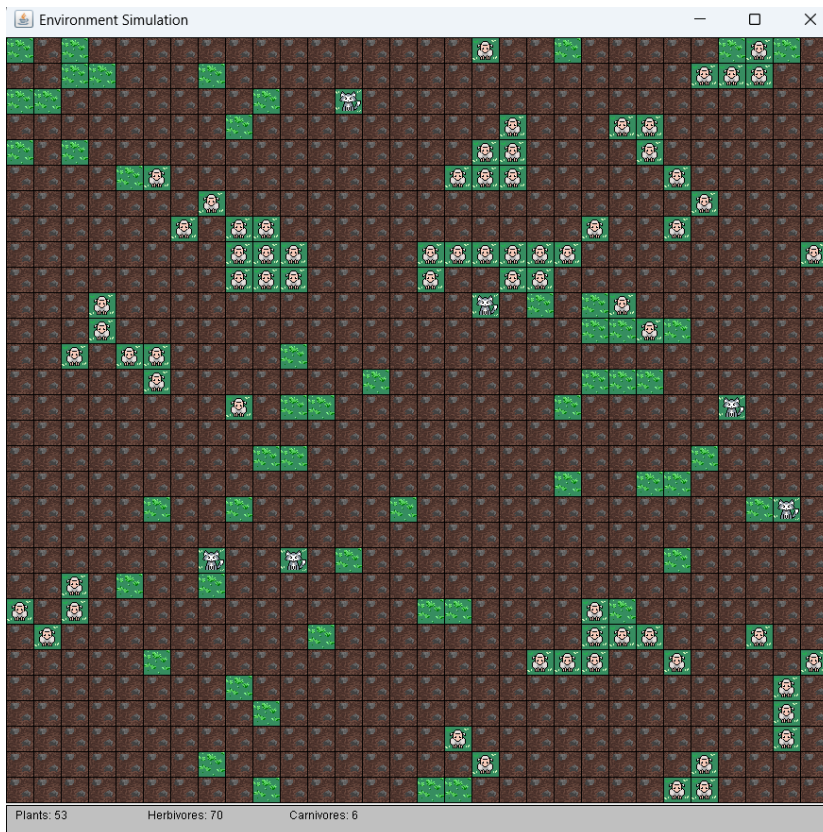
Herbivore Energy Threshold: 150

Herbivore Energy Decay: 4

Herbivore Default Energy: 150

Start Simulation

Cuối cùng sau khi lựa chọn xong thì người dùng sẽ nhìn thấy một lưới mô phỏng chuỗi thức ăn



3.2. Đánh giá kết quả

Chương trình mô phỏng hệ sinh thái phát triển dựa trên các kiến thức được giảng dạy trên lớp và tự tìm hiểu nên vẫn còn nhiều thiếu sót và hạn chế ,một phần cũng là do các thành viên trong nhóm còn thiếu nhiều kinh nghiệm trong việc xây dựng và phát triển ứng dụng.

Sau quá trình xây dựng và phát triển nhóm chúng em xin đưa ra một vài kết luận sau đây:

Ưu điểm:

- Chương trình đáp ứng đầy đủ các yêu cầu cơ bản của bài toán.

Nhược điểm

- Giao diện có thể không thân thiện hoặc hấp dẫn đối với người dùng phổ thông.

TÀI LIỆU THAM KHẢO

[1] Bài giảng học phần Lập trình hướng đối tượng của cô Nguyễn Thị Thu Trang.

[4] Lập trình hướng đối tượng bằng Java – Đoàn Văn Ban – NXB Khoa học và Kỹ Thuật

PHÂN CHIA CÔNG VIỆC

Hồ Đức Tú 20225676(TL)

- ✓ Class Animal 100%
- ✓ Class herbivore carnivore 50%
- ✓ GUI 100%
- ✓ Class Environment 50%
- ✓ Use case diagram

Vũ Minh Trí 20225940

- ✓ Class herbivore 50%
- ✓ Side 100%

- ✓ Reprort 80%

- ✓ Class diagram

Nguyễn Đức Trường 20225675

- ✓ Environment 50%

Nguyễn Anh Trà 20215151

- ✓ Class carnivore 50%

- ✓ Class environment 25%

Trần Văn Toàn 20225937

- ✓ Method move

- ✓ Report 20%