

Báo Cáo Assignment 2

I. Thông tin cá nhân

Họ và tên: Trương Huy Hoàng

Lớp: K55CA

Ngày sinh: 26/09/1992

Mã sinh viên: 10020148

II. Hướng dẫn cách biên dịch và chạy chương trình.

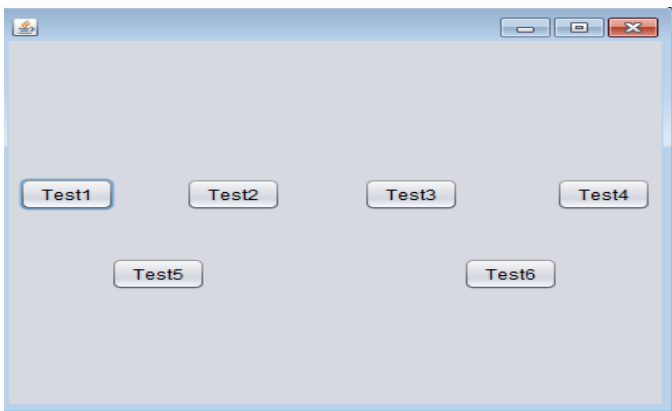
Biên dịch trên CMD(Terminal):

- Trỏ đến thư mục asg02
- Chạy CMD(Windows) or Terminal(Unix/Linux)
- Biên dịch và chạy với 2 lệnh:

```
javac -encoding UTF-8 Test.java
```

```
java Test
```

Khi màn hình hiển thị ra bảng thứ tự 6 Test (như hình vẽ). Để chạy Test nào ta nhấn vào test đó. **(Lưu ý chạy từng test một, xong test nào thì thoát luôn ra, sau đó vào lại bằng dòng lệnh “java Test” hoặc kích vào file “Test.bat”).**



- Cách thứ 2: nhấn file Test.bat (2 câu lệnh trên đã lưu trong file này).

III. Thuật toán các mức 2, 3 cho Test 3, 4.

Ta sẽ sử dụng thuật toán: Tìm đường đi ngắn nhất hoặc dài nhất từ một vị trí tới một mục tiêu xác định (kẻ thù hoặc thức ăn).

Một số quy tắc đã được kiểm chứng và đưa vào bài toán:

- Đường đi ngắn nhất: Ở đây là số bước đi từ vị trí hiện tại tới vị trí cần tới. Quãng đường này được tính bằng biểu thức: Giả sử điểm hiện tại là $thisPoint(x, y)$; Điểm mục tiêu cần tới là $desPoint(_x, _y)$. Quãng đường sẽ được tính bằng **Giá trị lớn nhất của hiệu 2 hoành độ và hiệu 2 tung độ** (chú ý lấy giá trị tuyệt đối của 2 hiệu này).
- + Với mỗi động vật, việc đầu tiên là tìm và xác định kẻ thù trong tầm nhìn của chúng. Nếu phát hiện kẻ thù nó lập tức di chuyển ra xa kẻ thù nhất mà không quan tâm thức ăn ở đâu. Động vật sẽ di chuyển ra xa bằng cách: **Tìm 1 trong 8 ô kề nó mà nó có thể đi tới, và xét tìm 1 ô có khoảng cách xa nhất so với kẻ thù, mà nó có thể đi tới.**
- + Nếu không tìm thấy kẻ thù, chúng sẽ tìm vị trí thức ăn gần nhất trong tầm nhìn của chúng. Nếu phát hiện có thức ăn thì nó sẽ đi theo đường đi ngắn nhất tới vị trí có thức ăn bằng cách: **Rà soát 8 ô xung quanh nó và tìm vị trí nó có thể đi tới mà gần nhất so với vị trí của thức ăn.**

Sau mỗi lần đủ khả năng di chuyển (đủ thời gian `moveDelay` để di chuyển sang 1 ô khác), nó sẽ lặp lại quá trình trên để đảm bảo rằng khi có thức ăn mới hoặc kẻ thù mới thì nó sẽ cập nhật và phân tích để có đường đi đúng đắn, để đảm bảo thuật toán tìm kiếm này là chính xác.

IV. Báo cáo kiểm thử.

STT	Mục đích thử nghiệm	Mô tả	Hướng dẫn chạy test	Kết quả trông đợi	Kết quả thu được	Kết luận
1	Hành vi của cỏ và hàng rào	Trên đồng cỏ, ngoài vòng hàng rào bao quanh, còn có một vòng hàng rào khép kín ở giữa là nhà của người nông dân. Bên ngoài vòng hàng rào trong, có một vài đám cỏ.	Test1	Cỏ sinh sôi theo đúng nhịp, lan rộng dần chiếm toàn bộ diện tích đồng cỏ nhưng không lan vào nhà của người nông dân	Như kết quả trông đợi	Đạt
2	Hành vi di chuyển và chết đói của cừu	Trên đồng cỏ có một vài con cừu ở bên trong một vòng hàng rào quây kín (chuồng cừu) cỏ sinh sôi ở bên ngoài nhưng không có ở trong chuồng và cũng không lan vào trong chuồng cừu.	Test2	Cừu di chuyển theo nhịp nhưng không ra được khỏi chuồng. Sau một thời gian cừu chết đói hết.	Như kết quả trông đợi	Đạt
3	Hành vi di chuyển tìm thức ăn của cừu	Trên đồng cỏ có cừu, cỏ và hàng rào làm chướng ngại vật. Các loài thực hiện đủ các hành vi sống của loài mình. Cấu hình thích hợp để hệ sinh thái cân bằng (không có loài nào có xu hướng diệt chủng)	Test3	Mức 2: cừu biết di chuyển về phía thức ăn nếu nhìn thấy	Như kết quả trông đợi	Đạt
4	Hành vi di chuyển tìm thức ăn của cừu và sói	Trên đồng cỏ có sói, cừu, cỏ và hàng rào làm chướng ngại vật. Các loài thực hiện đủ các hành vi sống của loài mình. Cấu hình thích hợp để hệ sinh thái cân bằng	Test4	Mức 3: cừu biết tránh sói nếu nhìn thấy.	Như kết quả trông đợi	Đạt
5	Bổ sung cho thấy rõ việc di chuyển tìm thức ăn của con cừu.	Trên đồng cỏ có cừu, cỏ và hàng rào làm chướng ngại vật. Các loài thực hiện đủ các hành vi sống của loài mình.	Test5	Mức 2: cừu biết di chuyển theo đường ngắn nhất về phía thức ăn nếu nhìn thấy	Như kết quả trông đợi	Đạt
6	Bổ sung cho thấy rõ việc di chuyển tìm thức ăn của con sói, và tránh sói của cừu	Trên đồng cỏ có sói, cừu và hàng rào làm chướng ngại vật. Các loài thực hiện đủ các hành vi sống của loài mình.	Test6	Mức 3: cừu biết tránh sói nếu nhìn thấy, và sói cũng đuổi cừu nếu nhìn thấy.	Như kết quả trông đợi	Đạt

V. Giải thích một số method trong class Animal

1. **radaForNearestFood(): Point:** Method này sẽ trả về một điểm trên đồng cỏ mà tại đó có thức ăn gần nhất trong phạm vi tầm nhìn.
2. **wayGoToTheNearestFood():** Point: Method này trả về một điểm trên đồng cỏ mà đó là con đường đi ngắn nhất tới nguồn thức ăn.
3. **radaForNearestEnemy(): ():** Point: Method này sẽ trả về một điểm trên đồng cỏ mà tại đó có kẻ thù gần nhất trong phạm vi tầm nhìn.
4. **wayToAvoidEnemy(): Point:** Method này trả về một điểm trên đồng cỏ mà đó là con đường đi ra xa nhất kẻ thù.
5. **isFood(Point): boolean {abstract}:** Method này trả về dạng đúng/ sai để kiểm tra xem tại điểm Point nó có tồn tại Entity là thức ăn của 1 Animal
6. **isEnemy(Point): boolean {abstract}:** Method này trả về dạng đúng/ sai để kiểm tra xem tại điểm Point nó có tồn tại Entity là kẻ thù của 1 Animal.
- (Hai method **isFood(Point)** và **isEnemy(Point)** sẽ thực hiện tại mỗi Animal **cụ thể** như Sheep, Wolf.).

7. **move(): void:** Method này thực hiện việc di chuyển sau khi đã đủ khoảng moveDelay. Nó sẽ tìm kẻ thù gần nhất trong tầm nhìn để tránh, nếu không thấy kẻ thù thì nó sẽ tìm thức ăn gần nhất trong tầm nhìn để ăn. Nếu vẫn không thấy thức ăn thì nó sẽ random đi 1 điểm bất kì trong khả năng của nó.

Sơ đồ UML

