

BÀI TẬP LỚN NMLT 193

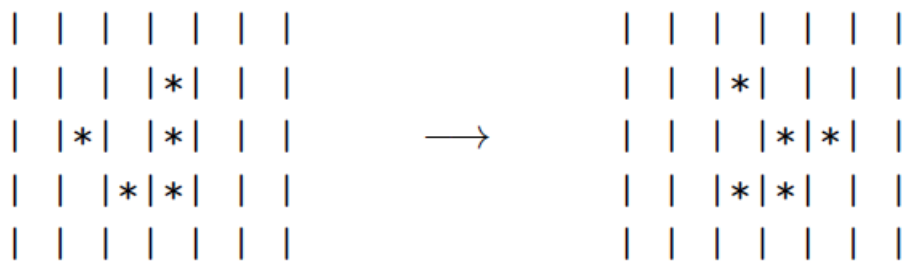
Phần 1: Tổng quan

Một trò chơi trong thế giới 2 chiều mở rộng vô hạn trong tất cả các hướng, chia vào các “cell”. Mỗi cell là “sống” hoặc “chết” tại một “thế hệ” được cho. Trò chơi gồm có một tập các luật thể hiện cách thức các cell tiến hóa qua các thế hệ nối tiếp nhau. Các luật này tính trạng thái của một cell ở thế hệ kế tiếp như một hàm của các trạng thái của các cell hàng xóm của nó trong thế hệ hiện tại. Mỗi cell tương tác với tám cell lân cận của nó, đó là các cell nằm cạnh theo chiều ngang, chiều dọc hoặc đường chéo. Tại mỗi bước theo thời gian, các chuyển đổi sau xảy ra:

1. Một cell sống có ít hơn hai người hàng xóm sống thì chết.
2. Một cell sống có hơn ba người hàng xóm sống cũng chết theo.
3. Một cell sống có đúng hai hoặc ba người hàng xóm thì sống.
4. Một cell chết với chính xác ba người hàng xóm sống trở nên sống.

Trong bài tập này, các SV sẽ cài đặt trò chơi với giới hạn là thế giới 2-D là hữu hạn. Các hàng xóm của một cell mà nằm phía ngoài cạnh thế giới được giả sử là chết.

Thí dụ: Thế giới hiện tại và thế giới sau một thế hệ kế tiếp



Ký hiệu: * là sống, _ là chết

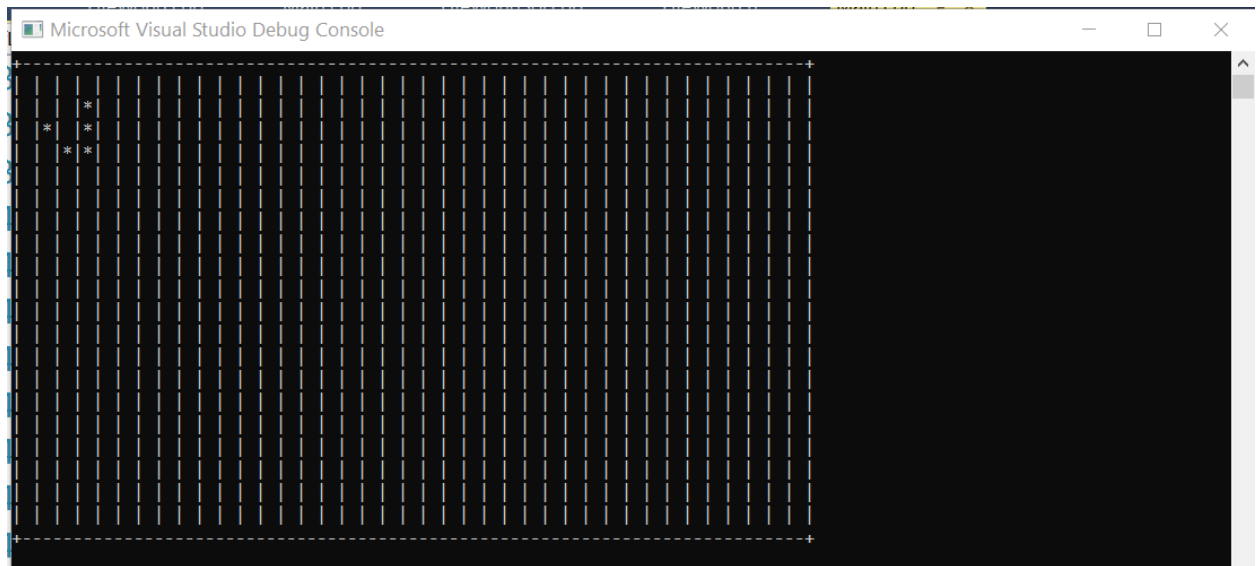
Phần 2: Hiện thực

Trong phần này, SV sẽ cài đặt các luật đã mô tả cho quá trình tiến hóa thế giới từ một thế hệ đến thế hệ kế tiếp. Một số hàm mẫu đã được cài đặt mục đích hỗ trợ sinh viên hiểu và sử dụng trong lúc hiện thực nằm ở 2 file “LifeWorld.h” và “LifeWorld.c”. SV nên tham khảo lại slide bài giảng (chapter 6 – Function) để hiểu cấu trúc tổ chức và cách tạo các file này.

SV cần hiện thực code của mình ở các phần có từ **//TODO:**

- a) Hiện thực hàm `hienthi_thegioi()` trong file “LifeWorld.c” để vẽ lên thế giới bằng cách sử dụng 2 tham số `WORLDWIDTH`, `WORLDHEIGHT`.

Minh họa thế giới sau khi đã thực thi hàm `khoitao_thegioi()`:



- b) Trong hàm `main`
- Khởi tạo thế giới trò chơi
 - Bằng cách nào để xuất ra trạng thái cuối cùng của thế giới khi đi qua tất cả bước tiến hóa.
- c) Hàm `main` gọi hàm `ketiep_thehe()` cho mỗi thế hệ để xử lý tiến trình tiến hóa. SV cần viết code đặt trạng thái mỗi cell trong thế hệ kế tiếp theo các luật đã nêu ở phần tổng quan. Khi trạng thái của tất cả các cell đã đặt cho thế hệ kế tiếp, gọi

hàm `cuoicung_tienhoa()` sẽ đặt trạng thái thế giới hiện tại đến trạng thái thế hệ kế tiếp và reset thế hệ kế tiếp

- d) Viết code cho hàm `lay_ketiep_trangthai(x,y)`, hàm này trả lại trạng thái kế tiếp (Sống hoặc Chết) của cell tại tọa độ cụ thể bằng cách dùng số hàng xóm sống (lấy được từ hàm `soluong_hangxom(x,y)`) và các luật trong trò chơi.
- e) Hoàn thành hàm `soluong_hangxom(x,y)`, hàm này trả lại số lượng các hàng xóm (8 cell lân cận) trạng thái là sống từ 1 cell cụ thể. Bởi vì thế giới là hữu hạn nên các cell kề mà nằm bên ngoài thế giới được giả sử là Chết.

Để giúp bạn kiểm tra mã của mình, chúng tôi đã cung cấp một số file kiểm tra: `pattern1.txt` (đầu ra của bạn phải khớp với “Phần 1” nếu qua 1 thế hệ kế tiếp) và `pattern2.txt` (đầu ra trong `pattern2out.txt`).

Phần 3: Nộp bài.

Để kết thúc, hãy viết một báo cáo thí nghiệm ngắn gọn (tối đa 2 trang):

- Mô tả trải nghiệm của bạn khi hoàn thành bài tập này bao gồm hình ảnh minh họa chạy dữ liệu từ 2 file mẫu đã cung cấp,
- Những thách thức bạn phải đối mặt và cách bạn giải quyết chúng cũng như những gì bạn học được từ bài tập này.
- Nộp file zip chứa tất cả mã của bạn (`main.c`, `LifeWorld.h` và `LifeWorld.c`) và báo cáo bài tập lớn của bạn, sử dụng hệ thống gửi trực tuyến trên BKEL.
- Định dạng: `Nhom-NMLT-04RZ03.zip`
 - o Minh họa: `1-NMLT-04RZ03.zip`

Deadline: 6/9/2020

Phần 4: Đánh giá và xử lý gian lận.

Tất cả có 5 câu nên mỗi câu đúng đạt 2 điểm.

Bài tập lớn phải được sinh viên (nhóm) TỰ LÀM. Sinh viên (nhóm) sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa các bài thu hoạch (nhất là phần kiến thức chuẩn bị). Trong trường hợp này, TẤT CẢ các bài nộp có sự giống nhau đều bị coi là gian lận. Do vậy sinh viên (nhóm) phải bảo vệ bài làm của mình.
- Sinh viên (nhóm) không hiểu bài làm do chính mình viết. Sinh viên (nhóm) có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những gì mình viết.

Bài bị phát hiện gian lận thì sinh viên (nhóm) sẽ nhận **0 điểm**.

Xin chúc mừng, bạn đã hoàn thành!