



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
MÔN: **ĐỒ HỌA MÁY TÍNH**
LỚP: 16CNTN

LAB 02

THUẬT TOÁN TÔ MÀU

1612842 LÊ THÀNH CÔNG

GVLT: TRẦN THÁI SON

GVTH: VÕ HOÀI VIỆT

Mục lục

I. Tổng quan.....	3
II. Nội dung.....	3
1. Thuật toán tô màu Scanline	3
a. Tô màu cho đa giác	3
b. Tô màu cho elip và hình tròn	5
2. Thuật toán tô màu boundary fill	7
3. Thuật toán tô màu boundary fill cải tiến	9
4. So sánh thực nghiệm các thuật toán	11
III. Nguồn tham khảo.....	12

I. Tổng quan

Mức độ hoàn thành của bài tập

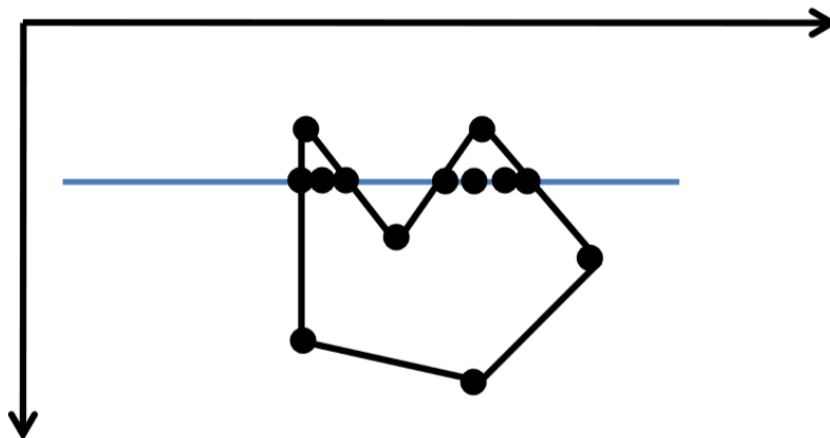
STT	Yêu cầu bài tập		Mức độ hoàn thành
1	Yêu cầu 1	Cài đặt thuật toán tô màu scanline và boundary fill	100%
2	Yêu cầu 2	Cải tiến thuật toán boundary fill bằng cách khử đệ quy. So sánh kết quả các thuật toán	100%
3	Yêu cầu 3	Sử dụng GUI để minh họa các thuật toán tô màu cho đa giác, hình tròn, đa giác. Cho phép người dùng chọn hình vẽ, nét vẽ, màu tô	100%

II. Nội dung

1. Thuật toán tô màu Scanline

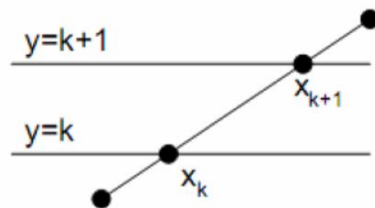
a. Tô màu cho đa giác

- Đây là thuật toán dòng quét (quét theo từng dòng) dùng để tô màu các đa giác lõm, đa giác tự cắt.
- Với mỗi dòng ta quét, ta cần xác định giao điểm giữa dòng quét và các cạnh của đa giác rồi tô màu các pixel thuộc đoạn giao đó.
- Để xác định các đoạn giao cần tìm các giao điểm của dòng quét và các cạnh của đa giác, sau đó sắp xếp tăng dần theo hoành độ.

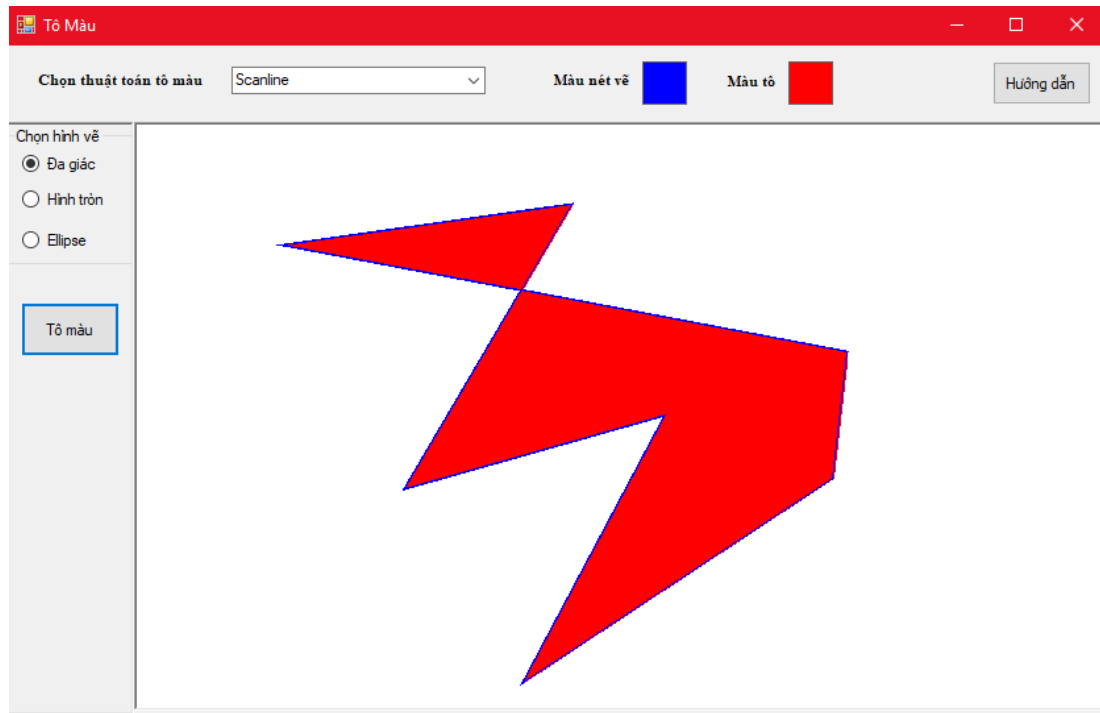


- Các bước thực hiện thuật toán:
 - Tìm y_{min} , y_{max} , x_{min} , x_{max} lần lượt là giá trị nhỏ nhất, lớn nhất của tung độ và hoành độ
 - Ta xét các dòng quét từ y_{min} đến y_{max} :
 - Tìm tất cả hoành độ giao điểm của dòng quét với các cạnh đa giác
 - Sắp xếp các hoành độ giao điểm theo thứ tự tăng dần x_0, x_1, x_2, \dots
 - Tô màu các đoạn thẳng theo từng cặp $(x_0, x_1), (x_2, x_3), \dots$
- Vấn đề gặp phải đó là việc tìm giao điểm khá phức tạp nếu giải phương trình giao điểm làm chậm thuật toán. Hơn nữa, không phải lúc nào cũng có cạnh cắt dòng quét. Đôi khi dòng quét qua các đỉnh đa giác làm ảnh hưởng đến việc nhóm từng cặp $(x_0, x_1), (x_1, x_2), \dots$ để tô hoặc trường hợp đường thẳng nằm ngang cũng cần xử lý.
- Giải pháp:
 - Tăng tốc việc tìm kiếm giao điểm sử dụng kết quả trước nhờ hệ số góc

$$x_{k+1} - x_k = \frac{1}{m}((k+1) - k) = \frac{1}{m} \text{ hay } x_{k+1} = x_k + \frac{1}{m}.$$



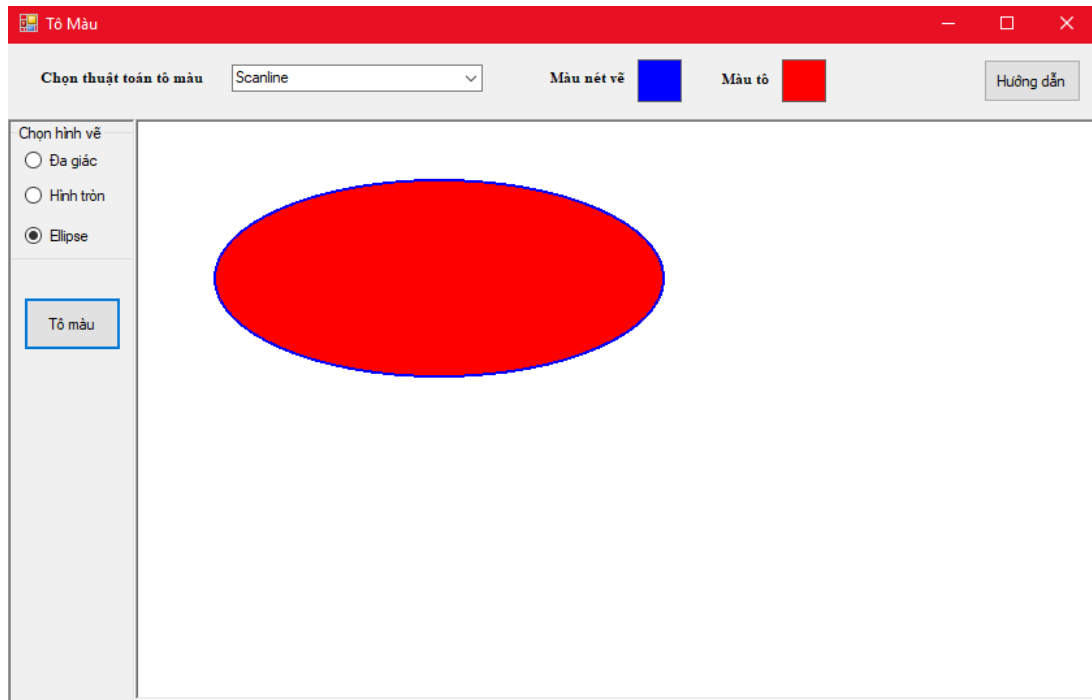
- Sử dụng quy tắc Odd-even tô màu cho các trường hợp số giao điểm dòng quét với đa giác là lẻ.
- Kết quả test:



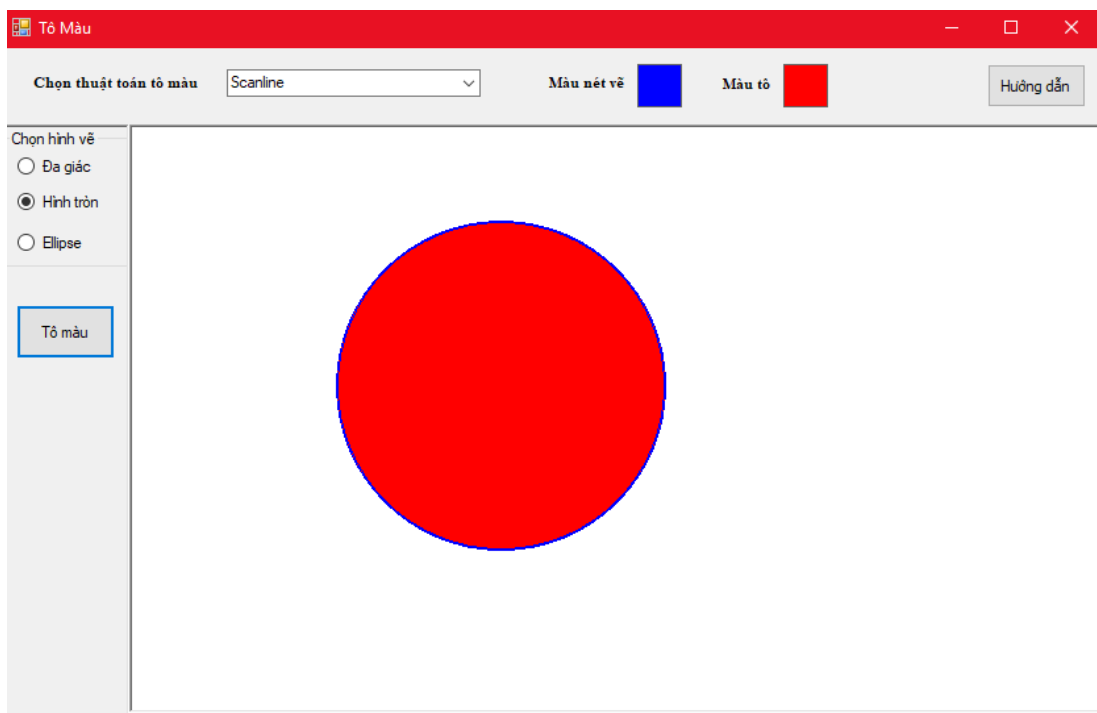
Hình 1. Đa giác - Scanline

b. Tô màu cho elip và hình tròn

- Ý tưởng vẫn là quét từng dòng sau đó tìm giao điểm của dòng quét và ellipse/đường tròn
- Ellipse/đường tròn quy định lưu trữ như một hình chữ nhật. Có điểm góc trên bên trái, chiều dài, chiều rộng. Bán trục lớn $a = \text{rộng}/2$, bán trục nhỏ $b = \text{rộng}/2$
- Để tìm giao điểm ta áp dụng công thức của Ellipse: $\frac{(x^2-x_0)^2}{a^2} + \frac{(y^2-y_0)^2}{b^2} = 1$. Với y là tung độ dòng đang quét, (x_0, y_0) là tâm của ellipse hiện tại. Ta tìm được x theo các giá trị còn lại của công thức. Đường tròn là trường hợp đặc biệt của Ellipse nên vẫn dùng được công thức trên.
- Với x mang 2 giá trị x_1, x_2 vừa tìm được, ta tô màu đoạn thẳng từ x_1 đến x_2
- Tiếp tục các dòng quét còn lại từ y_{\min} đến y_{\max}
- Kết quả test:



Hình 2. Elip - Scanline

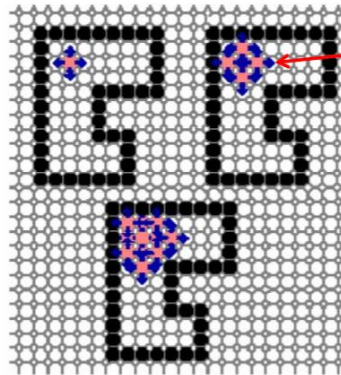


Hình 3. Hình tròn - Scanline

2. Thuật toán tô màu boundary fill

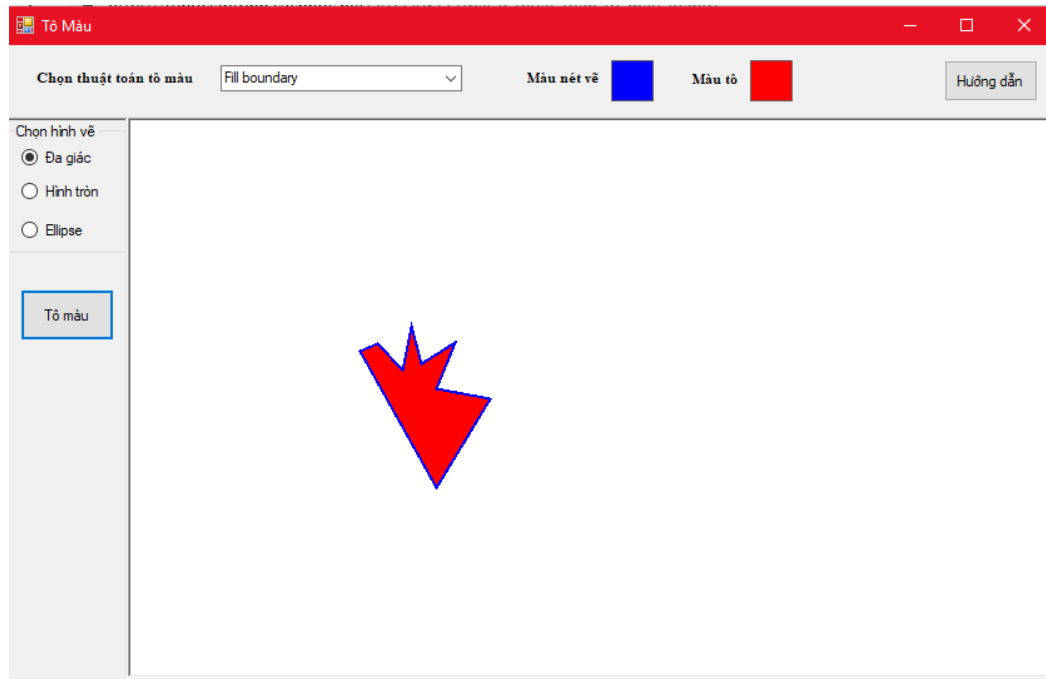
- Đây là thuật toán tô màu loang hay còn gọi là tô màu theo đường biên, tô lân cận. Thuật toán này thực hiện đệ quy tô màu bắt đầu từ một điểm nằm trong vùng cần tô, kiểm tra điểm đó khác màu với màu cần tô và màu của đường biên hay không. Nếu khác màu thì thực hiện tô màu và đệ quy đến vị trí lân cận, tiếp tục kiểm tra và đệ quy tiếp đến khi gặp biên và không còn chỗ để tô nữa thì dừng.

Boundary-Fill Algorithm

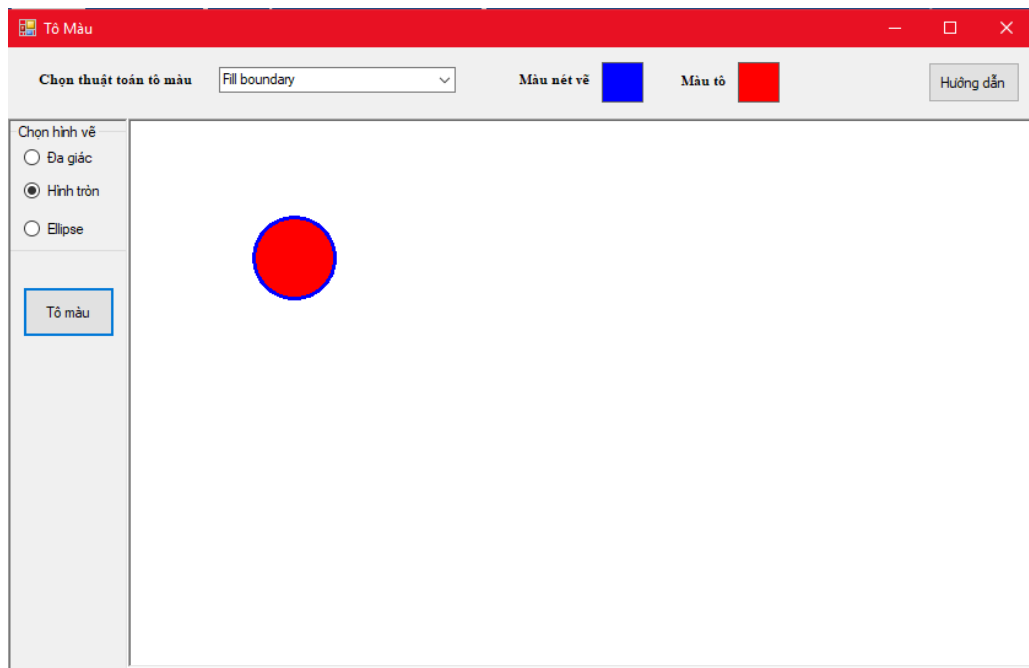


1. Start at one point inside polygon.
2. Paint the interior outward toward the boundary by using connected part of the starting point.

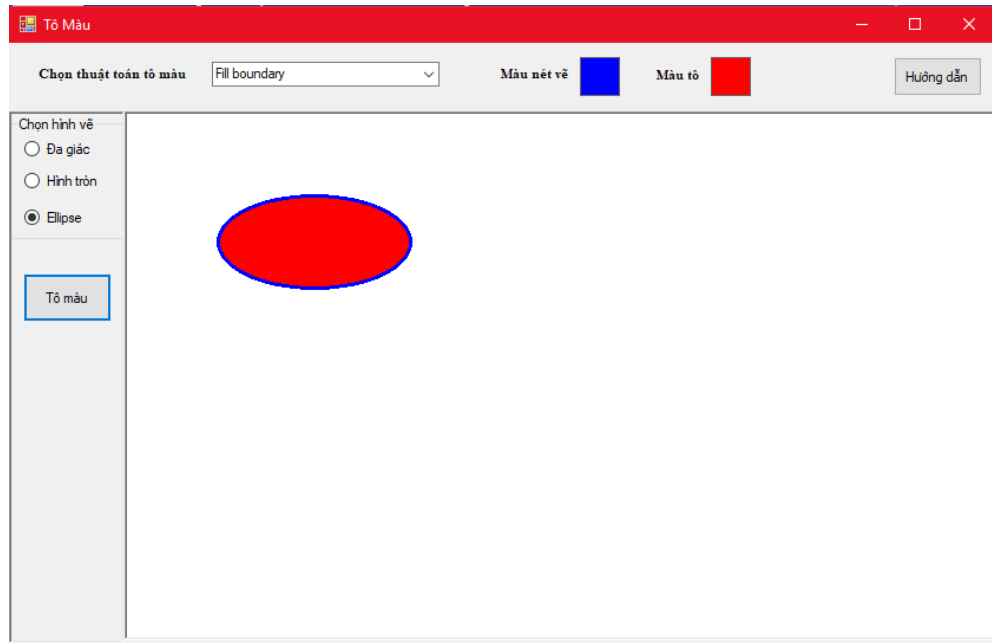
- Các bước thuật toán đệ quy:
 - o Chọn vùng cần tô nằm trong hình có đường biên
 - o Xác định chọn điểm (x,y) trong vùng đó
 - o Tô điểm (x,y) và loang lân cận
- Ưu điểm:
 - o Có thể tô vùng có hình dạng bất kì miễn sao có đường biên giới hạn
 - o Cài đặt thuật toán ngắn gọn dễ dàng
- Nhược điểm:
 - o Nếu tô vùng kích thước lớn sẽ bị tràn bộ nhớ (Stackoverflow)
 - o Việc gọi đệ quy đến 4 điểm lân cận không quan tâm đã xét hay chưa nên bị xét quá nhiều lần, không hiệu quả.
- Kết quả test:



Hình 4. Đa giác – Fill boundary



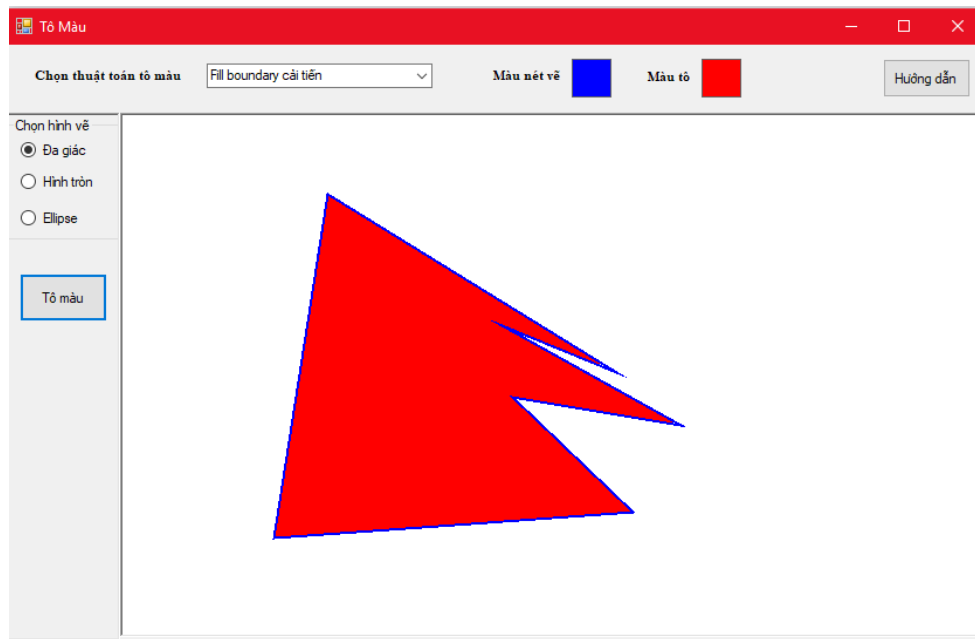
Hình 5. Hình tròn – Fill boundary



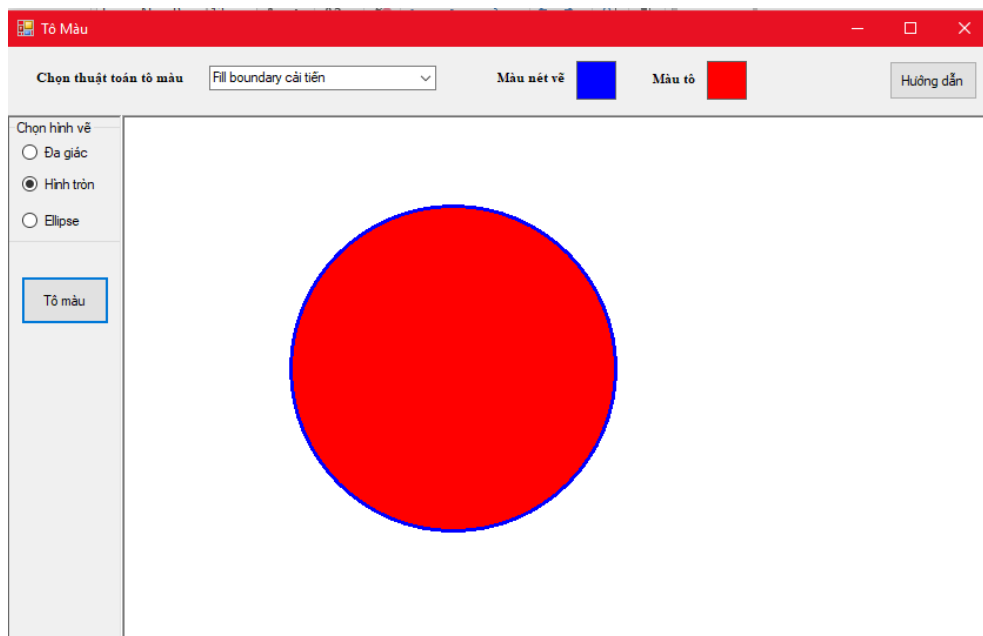
Hình 6. Elip – Fill boundary

3. Thuật toán tô màu boundary fill cải tiến

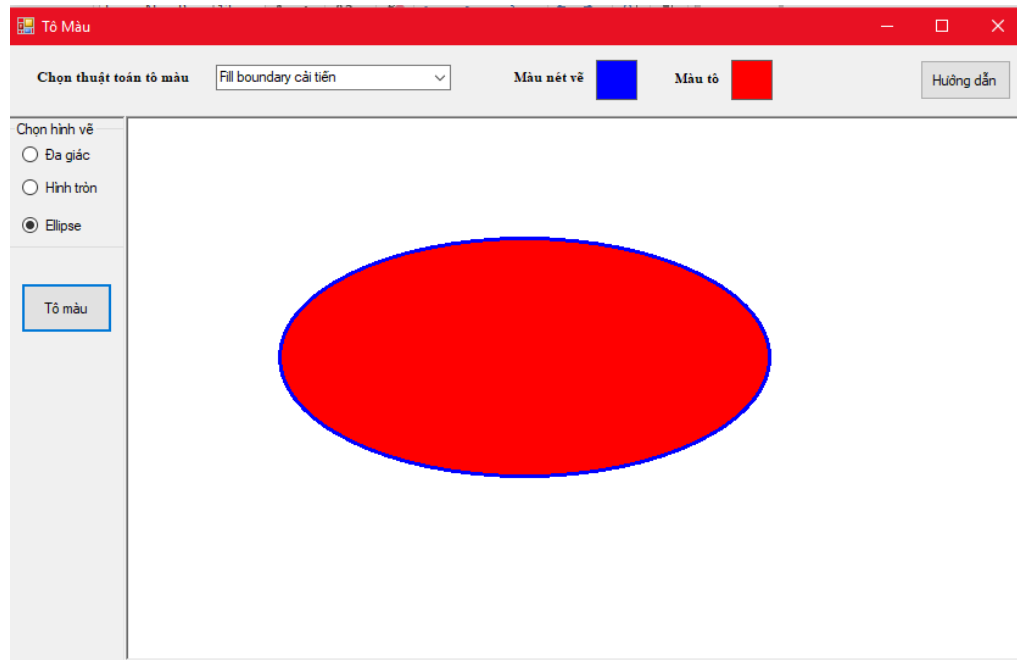
- Để khắc phục những vấn đề khi đệ quy, ta cần khử đệ quy.
- Chúng ta sẽ tiến hành loang dần thực hiện hàng đợi Queue để khử đệ quy
- Các bước thực hiện thuật toán:
 - Đưa điểm m đầu tiên (x,y) vào hàng đợi Queue
 - Lặp lại các bước sau nếu Queue chưa rỗng:
 - Pop ra 1 điểm ở đầu Queue
 - Lấy điểm m và xét các điểm lân cận $(x-1,y)$, $(x+1,y)$, $(x, y+1)$, $(x, y -1)$
 - Mỗi lần xét kiểm tra điểm xét có khác với màu tô và màu biên hay không? Nếu khác thì thực hiện tô và push điểm xét đó vào hàng đợi
 - Gán m = giá trị đầu tiên của hàng đợi.
- Thuật toán fill boundary phải chọn một điểm nằm trong vùng tô, nên nếu hình có biên bị khuyết, hoặc biên không khép kín thì thuật toán sẽ tô loang vô hạn. Nên cần ràng buộc trường hợp đó giới hạn lại (x, y) phải nằm trong khung vẽ khi loang.
- Kết quả test:



Hình 7. Đa giác – Fill boundary cải tiến



Hình 8. Hình tròn – Fill boundary cải tiến



Hình 9. Elip – Fill boundary cải tiến

4. So sánh thực nghiệm các thuật toán

- Sau quá trình test thực nghiệm, thuật toán scanline cho thời gian xử lý nhanh nhất, tiếp theo là fill boundary cải tiến và cuối cùng là fill boundary.
- Thuật toán scanline áp dụng được cho đa giác lồi, không lồi. Tuy nhiên khi sử dụng thuật toán này cần biết thông tin về các đỉnh của hình hoặc thông tin đặc trưng của hình đó như tọa độ, công thức hình học.
- Thuật toán fill boundary và fill boundary cải tiến áp dụng cho các hình đa giác lồi và các hình khác nếu có đường biên khép kín. Do chỉ dựa trên màu sắc đường biên nên không cần bất kỳ thông tin nào khác vẫn có thể tô được. Tuy nhiên, cần xác định đúng điểm bắt đầu loang nằm bên trong khu vực cần tô để có thể tô đúng được hình vẽ.
- Thuật toán fill boundary do sử dụng đệ quy nên nếu tô màu hình vẽ có diện tích lớn dẫn đến Stackoverflow làm crash chương trình, đây là điểm hạn chế chí mạng của thuật toán này.

III. Nguồn tham khảo

Slide bài giảng lý thuyết

<https://www.codeproject.com/Articles/1355/Professional-C-Graphics-with-GDI>

https://www.tutorialspoint.com/computer_graphics/polygon_filling_algorithm.htm

<https://www.geeksforgeeks.org/boundary-fill-algorithm/>

<https://www.thecrazyprogrammer.com/2017/02/boundary-fill-algorithm-in-c.html>

<http://mycurlycode.blogspot.com/2016/02/fill-polygon-using-scan-line-fill.html>

<https://tuhoclaptrinh.cachhoc.net/2017/03/19/bai-8-thuat-toan-to-mau-scanline/>

<https://tuhoclaptrinh.cachhoc.net/2017/03/22/bai-9-thuat-toan-to-mau-loang/>