**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

****

**BÁO CÁO ĐỒ ÁN**

**VẼ ĐỐI TƯỢNG 2D**

**Môn học: Đồ họa máy tính**

Sinh viên thực hiện:Dũ Quốc Huy – 20120101

Giáo viên hướng dẫn: Trần Thái Sơn

Võ Hoài Việt

Thành phố Hồ Chí Minh, ngày 21 tháng 10 năm 2022

Mục lục

[1. Tổng quan về thiết kế chương trình 3](#_Toc117301960)

[2. Phân tích thuật toán 3](#_Toc117301961)

[2.1. Vẽ đường thẳng bằng DDA 4](#_Toc117301962)

[2.1.1. Tham số đầu vào: 4](#_Toc117301963)

[2.1.2. Ý tưởng thuật toán 4](#_Toc117301964)

[2.1.3. Kết quả thực hiện: 5](#_Toc117301965)

[2.2. Vẽ đường thẳng bằng Bresenham 5](#_Toc117301966)

[2.2.1. Tham số đầu vào 5](#_Toc117301967)

[2.2.2. Ý tưởng thuật toán 5](#_Toc117301968)

[2.2.3. Kết quả thực hiện 7](#_Toc117301969)

[2.3. Vẽ dường tròn bằng MidPoint 8](#_Toc117301970)

[2.3.1. Tham số đầu vào 8](#_Toc117301971)

[2.3.2. Ý tưởng thuật toán 8](#_Toc117301972)

[2.3.3. Kết quả thực hiện 10](#_Toc117301973)

[2.4. Vẽ hình Ellipse bằng MidPoint 11](#_Toc117301974)

[2.4.1. Tham số đầu vào 11](#_Toc117301975)

[2.4.2. Ý tưởng thuật toán 11](#_Toc117301976)

[2.4.3. Kết quả thực hiện 14](#_Toc117301977)

[2.5. Vẽ Parabol bằng MidPoint 15](#_Toc117301978)

[2.5.1. Tham số đầu vào 15](#_Toc117301979)

[2.5.2. Ý tưởng thuật toán 15](#_Toc117301980)

[2.5.3. Kết quả thực hiện: 17](#_Toc117301981)

[2.6. Vẽ Hyperbol bằng MidPoint 18](#_Toc117301982)

[2.6.1. Tham số đầu vào 18](#_Toc117301983)

[2.6.2. Ý tưởng thuật toán 18](#_Toc117301984)

[2.6.3. Kết quả thực hiện 20](#_Toc117301985)

[3. Đánh giá thuật toán 21](#_Toc117301986)

[4. Hướng dẫn sử dụng 22](#_Toc117301987)

[5. Tài liệu tham khảo 23](#_Toc117301988)

1. Tổng quan về thiết kế chương trình

Nội dung đồ án: sử dụng thư viện đồ họa GLUT để vẽ các đối tượng 2D: đường thẳng, đường tròn, ellipse, đường parabol và đường hyperbol bằng các thuật toán trong chương trình lý thuyết (DDA, Bresenham, MidPoint).

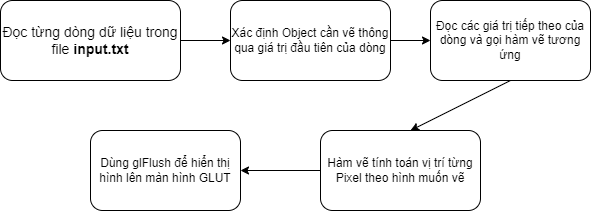
Đầu vào: chương trình thực hiện việc đọc đầu vào thông qua tập tin **input.txt**, với mỗi dòng là các tham số để vẽ hình các nhau bằng khoảng trắng với giá trị đầu tiên là loại đối tượng muốn vẽ, các giá trị sau là giá trị điều khiển của đối tượng được theo quy ước có sẵn.

Đầu ra: sau khi thực hiện việc chạy chương trình, sẽ xuất hiện màn hình GLUT là các đối tượng được vẽ, màn hình Console là thời gian vẽ của từng đối tượng tính tới milisecond.

Tổ chức mã nguồn:

* Mã nguồn sẽ bao gồm 1 tập tin Draw2Dbject.cpp có hàm main để xử lí việc hiển thị giao diện GLUT và hàm để xử lí việc đọc thông tin từ file **input.txt.**
* Nhiều tập tin có đuôi .h gồm: Header.h (Khai báo thư viện, khai báo các tên hàm để cho việc vẽ), SupportDraw.h (chứa các hàm hỗ trợ việc setPixel và tính thời gian thực hiện), LineDDA.h, CircleMidPoint.h,… (chứa hàm tính toán để vẽ từng pixel của hình). Ngoài ra còn có DrawByOpenGL.h (vẽ object dựa vào các hàm do OpenGL cung cấp)
* Tập tin **input.txt** để chứa dữ liệu đầu vào.

Quá trình vẽ 1 hình:



1. Phân tích thuật toán
   1. Vẽ đường thẳng bằng DDA
      1. Tham số đầu vào:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | x1 | y1 | x2 | y2 |

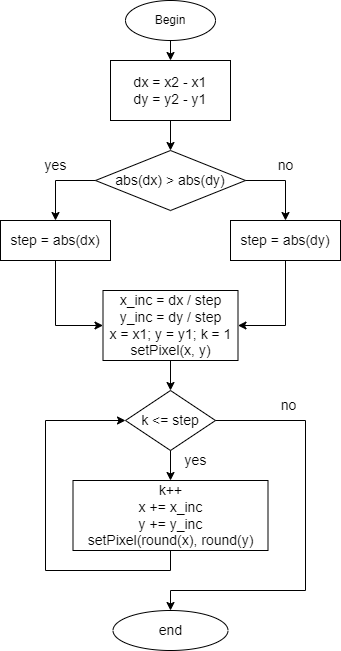
(x1, y1): tọa độ đỉnh đầu

(x2, y2): tọa độ đỉnh cuối

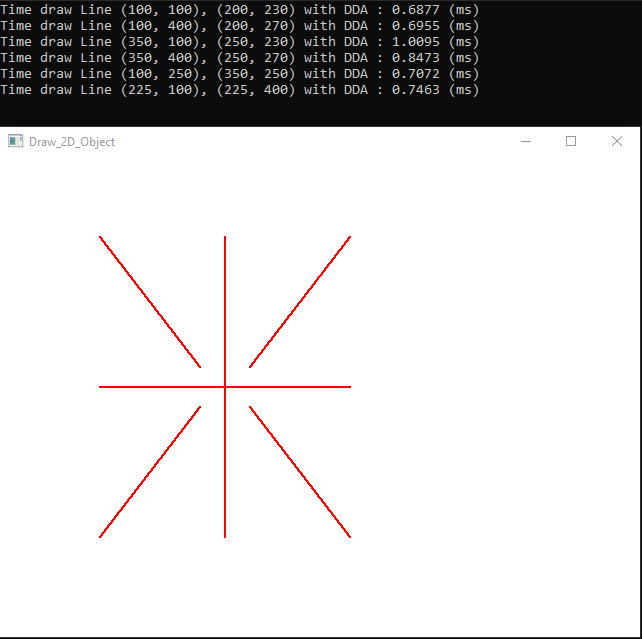
* + 1. Ý tưởng thuật toán

Trước tiên cần xác định được đường thẳng sẽ biến thiên theo x hay theo y nhiều hơn. Vì vẽ theo giá trị biến thiên nhiều hơn sẽ giúp vẽ được nhiều pixel hơn, hình vẽ sẽ mượt hơn. Biến step sẽ giúp lưu lại số lượng pixel sẽ vẽ, các biến x\_inc/ y\_inc (kiểu float) sẽ lưu lại việc tăng/ giảm của x/ y qua từng pixel. Sau đó làm tròn số sẽ giúp lấy chính xác giá trị nguyên của tọa độ pixel.

Lưu đồ thuật toán vẽ đường thẳng bằng DDA(Digital Differential Analyzer) như sau:



* + 1. Kết quả thực hiện:



* 1. Vẽ đường thẳng bằng Bresenham
     1. Tham số đầu vào

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | x1 | y1 | x2 | y2 |

(x1, y1): tọa độ đỉnh đầu

(x2, y2): tọa độ đỉnh cuối

* + 1. Ý tưởng thuật toán

Với trường hợp vẽ y theo x, với đường thẳng đi từ dưới lên trên:

Ảnh có chứa ăng-ten

Mô tả được tạo tự động

, 



So sánh giá trị của  và  :

 với 

Chia 2 vế cho 

 với 

Giảm việc tính toán của  trong (\*) như sau:





Xác định hằng số tăng ở mỗi bước





Xác định giá trị ban đầu:

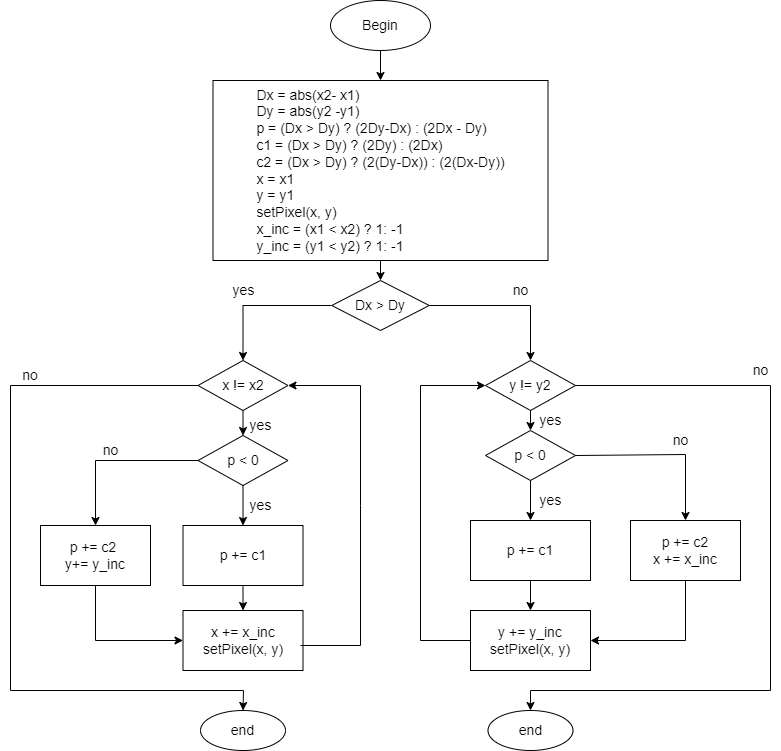




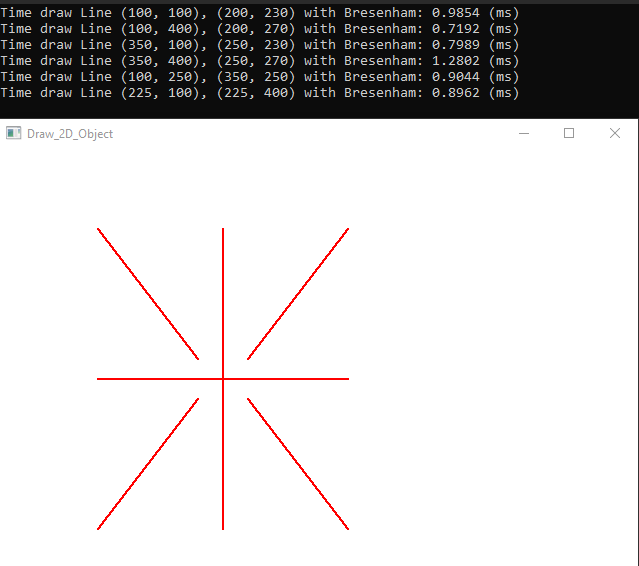


Tương tự chứng minh cho các trường hợp vẽ từ trên xuống, trường hợp vẽ x theo y.

Lưu đồ thuật toán vẽ đường thẳng bằng Bresenham đầy đủ các trường hợp:



* + 1. Kết quả thực hiện



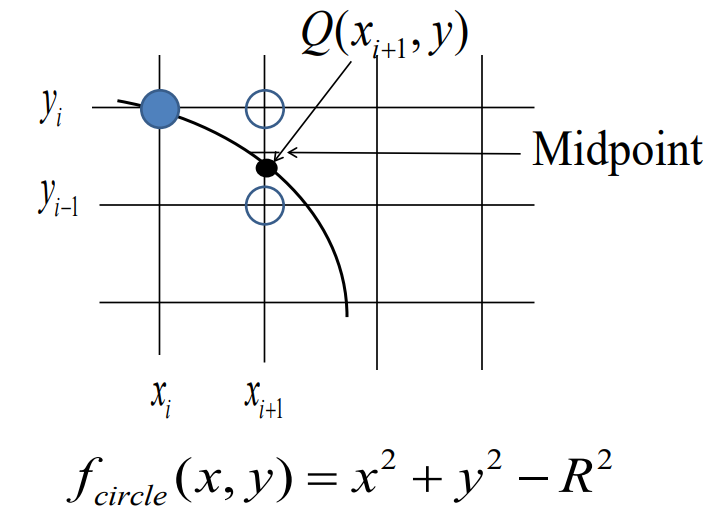
* 1. Vẽ dường tròn bằng MidPoint
     1. Tham số đầu vào

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | xT | yT | r |

(xT, yT): tọa độ tâm

r: bán kính

* + 1. Ý tưởng thuật toán





Giả sử  đã được định sẵn, ta cần xác định liệu pixel ở vị trí  hay  gần đường tròn hơn. Ta tính  như sau:

 (\*\*\*)

Nếu , điểm midpoint đó nằm trong đường tròn =>  gần hơn

Nếu , điểm midpoint đó nằm trên hoặc ngoài đường tròn =>  được chọn

Giảm việc tính toán của  của (\*\*\*) bằng cách





Và xác định mức tăng tương ứng cho mỗi bước

Nếu 

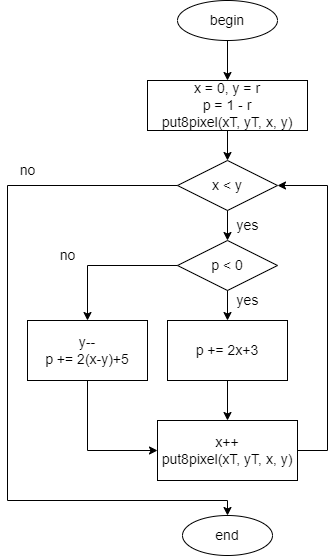
Nếu 

Xác định giá trị 



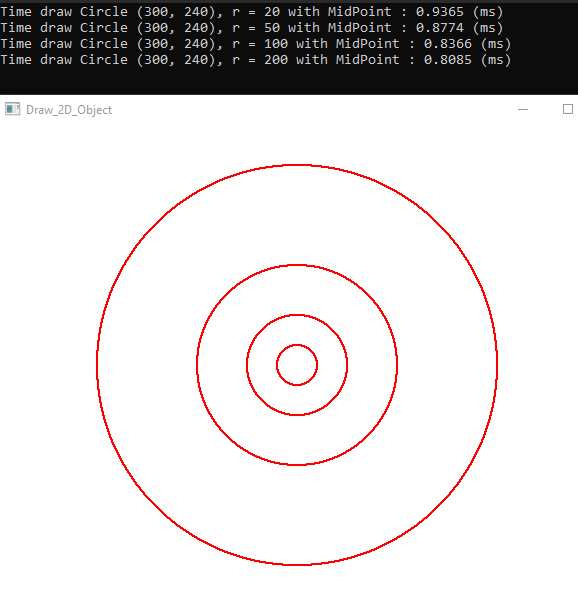


Lưu đồ vẽ hình tròn bằng thuật toán MidPoint:



Vì hình tròn có tính đối xứng, chỉ cần vẽ cung tròn từ 90 đến 45 độ, sử dụng hàm put8pixel để hoàn thiện các vị trí đối xứng khác.

* + 1. Kết quả thực hiện



* 1. Vẽ hình Ellipse bằng MidPoint
     1. Tham số đầu vào

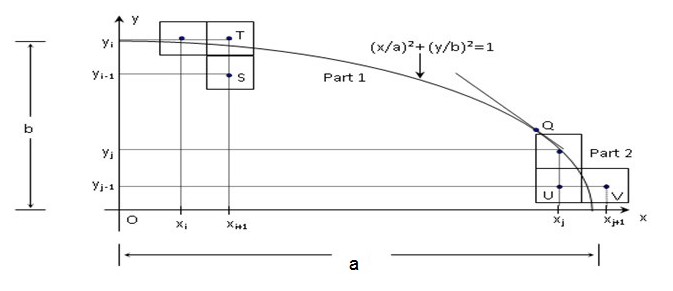
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | xT | yT | a | b |

(xT, yT): tọa độ tâm

a: ½ độ dài trục lớn

b: ½ độ dài trục nhỏ

* + 1. Ý tưởng thuật toán



Phương trình Ellipse: 

Ta chỉ cần vẽ ¼ hình ellipse sau đối lấy tối xứng 4 đổi để ra được hình hoàn chỉnh

Ở góc phần tư này, ta chia ra thành 2 giai đoạn: vẽ y theo x (từ 90 độ đến vị trí dx = dy), vẽ x theo y (từ vị trí dx=dy đến 0 dộ). 

* Giai đoạn 1: Vẽ y theo x:









Nếu , điểm midpoint nằm trong hình ellipse, 



Nếu , điểm midpoint nằm trong hình ellipse, 





* Giai đoạn 2: Vẽ x theo y









Nếu , điểm midpoint nằm ngoài hình ellipse, 



Nếu , điểm midpoint nằm trong hình ellipse, 





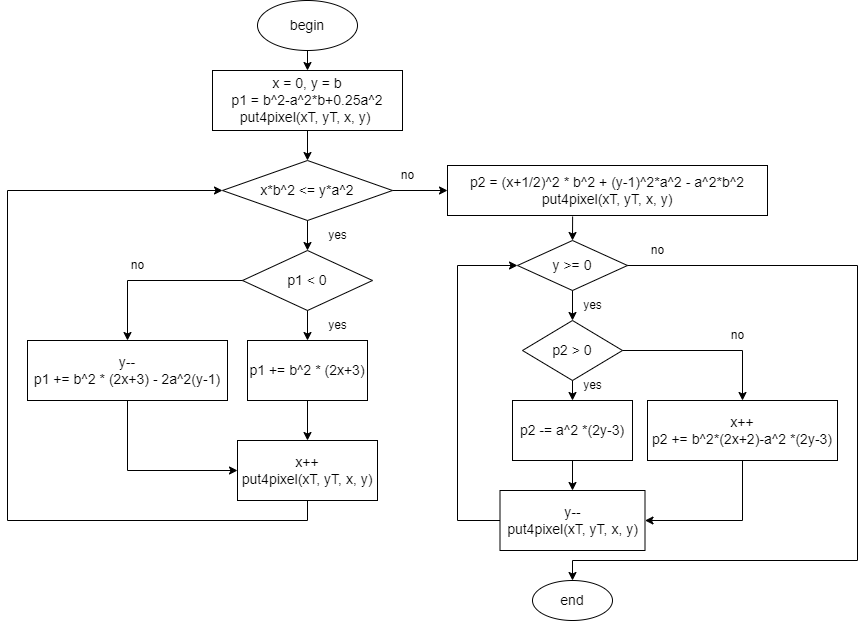
Xác định  trước khi bắt đầu vẽ y theo x, cho y = b, x = 0





Trước khi bắt đầu vẽ x theo y, ta sử dụng lại tọa độ (x, y) ở cuối giai đoạn trước để tính 

Lưu đồ vẽ hình Ellipse bằng thuật toán MidPoint



* + 1. Kết quả thực hiện

Ảnh có chứa văn bản, các môn điền kinh, thể thao

Mô tả được tạo tự động

* 1. Vẽ Parabol bằng MidPoint
     1. Tham số đầu vào

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | xT | yT | p |

(xT, yT): tọa độ đỉnh

p: khoảng cách từ đỉnh tới tiêu cự

* + 1. Ý tưởng thuật toán

Phương trình đường parabol: 

Ta chỉ cần vẽ một nữa parabol ở phần bên phải, sau đó lấy đối xứng qua trục đối xứng x sẽ ra được hình parabol hoàn chỉnh

Để vẽ phần bên phải, ta chia làm 2 giai đoạn: vẽ y theo x (từ đến điểm dx = dy), sau đó vẽ x theo y (từ điểm dx = dy đến vô cùng)

* Giai đoạn 1: Vẽ y theo x









Nếu , điểm midpoint nằm phía trên đường parabol, 



Nếu , điểm midpoint nằm phía dưới/ trên đường parabol, 



Xác định , với x = 0, y = 0



* Giai đoạn 2: Vẽ x theo y









Nếu , điểm midpoint nằm phía trên đường parabol, 

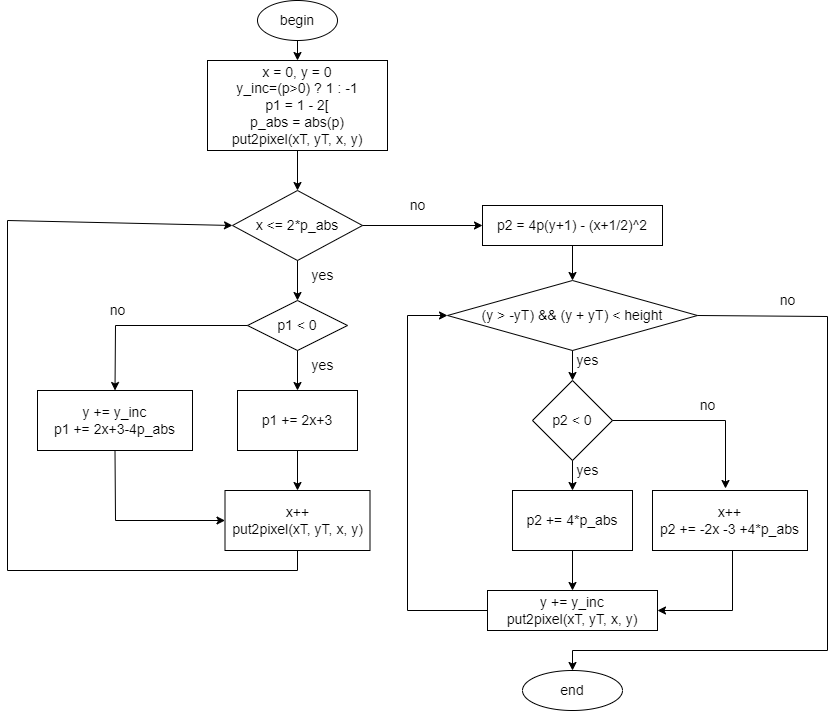


Nếu , điểm midpoint nằm phía dưới/ trên đường parabol, 



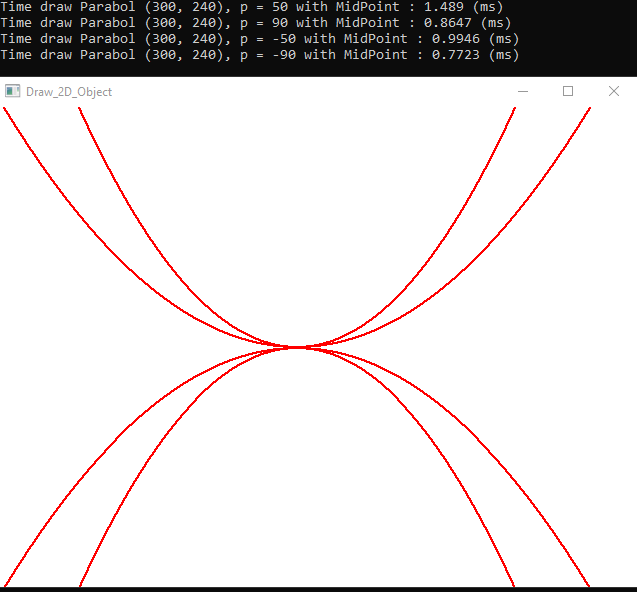
Điểm  của giai đoạn này là  với x, y là điểm cuối của giai đoạn 1

Lưu đồ vẽ hình Parabol bằng MidPoint:



Với trường hợp p < 0, parabol sẽ hướng xuống dưới.

* + 1. Kết quả thực hiện:



* 1. Vẽ Hyperbol bằng MidPoint
     1. Tham số đầu vào

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | xT | yT | a | b |

(xT, yT): tọa độ tâm

a: độ dài trục thực

b: độ dài trục ảo

* + 1. Ý tưởng thuật toán

Để tối ưu thời gian vẽ hình, ta sẽ chia 2 dạng hình hyperbol: a >= b và ngược lại.

Với trường hợp a >= b, thực hiện 2 giai đoạn: vẽ x theo y trước (từ tâm đến điểm dx = dy), sau đó vẽ y theo x (từ điểm dx = dy đến vô cùng)

Với trường hợp a < b thực hiện ngược lại so với trường hợp trên.

Phường trình đường Hyperbol: 



Với trường hợp a >= b, ta chỉ vẽ phần y dương, x dương sau đó lấy đối xứng để ra hình hoàn chỉnh (giống cách vẽ ellipse)

* Giai đoạn 1: vẽ x theo y









Nếu , điểm midpoint nằm phía dưới đường hyperbol, 



Nếu , điểm midpoint nằm phía trên đường hyperbol, 





Xác định , với x =a, y = 0





* Giai đoạn 2: vẽ y theo x









Nếu , điểm midpoint nằm phía trên đường hyperbol, 

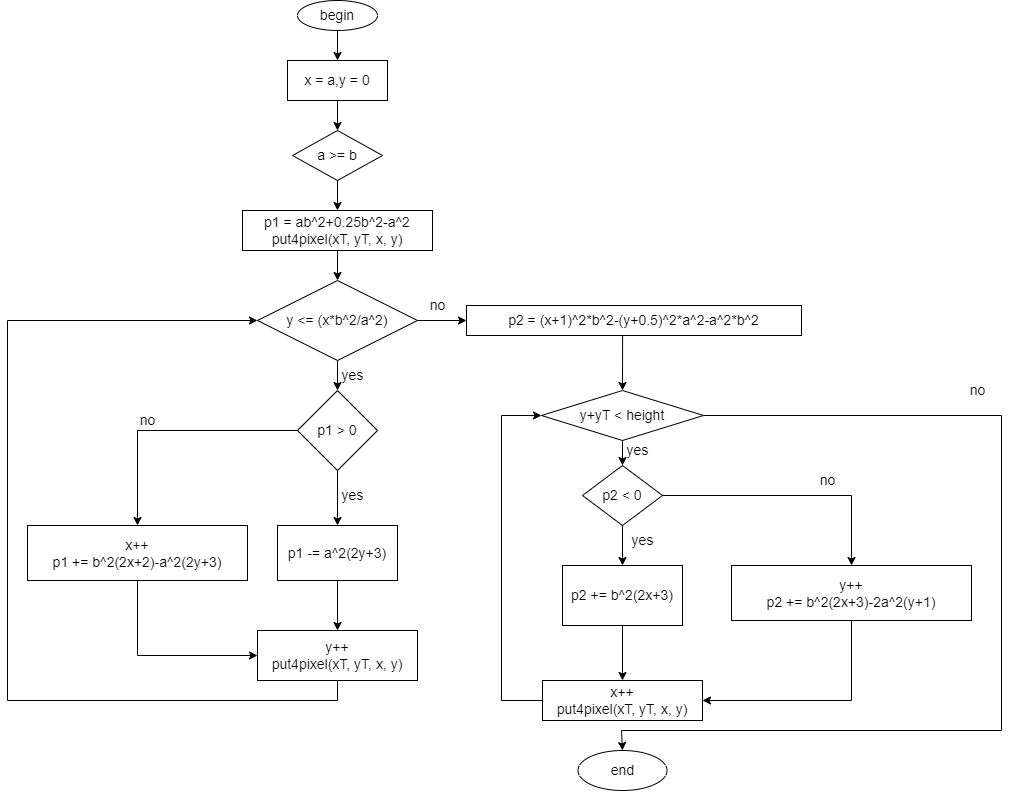


Nếu , điểm midpoint nằm phía dưới đường hyperbol, 



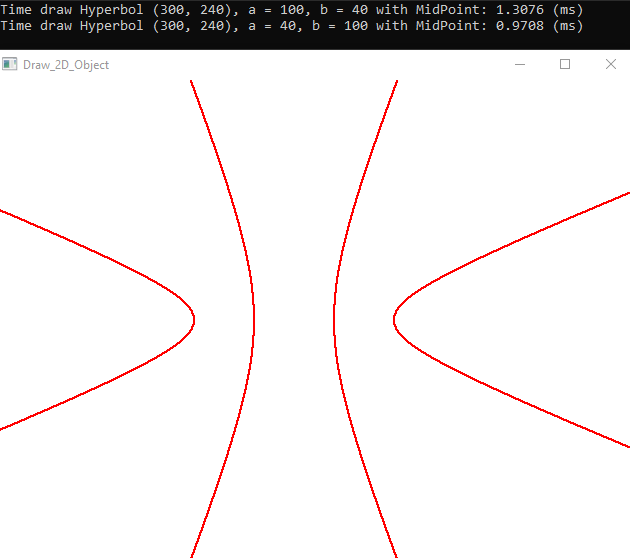
Điểm  ở giai đoạn này được tính với x, y là tọa độ cuối của giai đoạn 1.

Lưu đồ vẽ hinh Hyperbol bằng thuật toán MidPoint với a >= b:



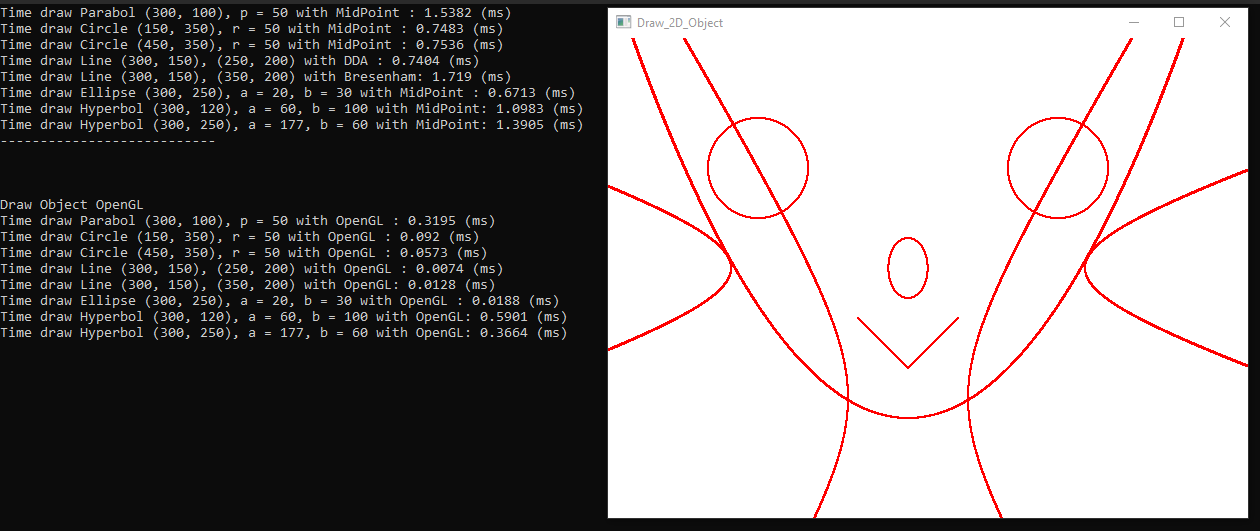
Việc vẽ cho trường hợp a < b, chỉ cần thay đổi thứ 2 vẽ (vẽ y theo x trước rồi vẽ x theo y).

* + 1. Kết quả thực hiện



1. Đánh giá thuật toán

Đây là kết quả thử vẽ một vài hình giữa các thuật toán tự cài đặt và việc xử dụng các hàm có sẵn trong OpenGL



Bảng so sánh thời gian (ms)

|  |  |  |
| --- | --- | --- |
| Thuật toán  Object | Thuật toán tự cài đặt | Sử dụng hàm OpenGL cung cấp |
| 4 300 100 50 | 1.5382 ms | 0.3195 ms |
| 2 150 350 50 | 0.7483 ms | 0.092 ms |
| 2 450 350 50 | 0.7536 ms | 0.0573 ms |
| 0 300 150 250 200 | 0.7404 ms | 0.0128 ms |
| 1 300 150 350 200 | 1.719 ms | 0.0128 ms |
| 3 300 250 20 30 | 0.6713 ms | 0.0188 ms |
| 5 300 120 60 100 | 1.0983 ms | 0.5901 ms |
| 5 300 250 177 60 | 1.3905 ms | 0.3664 ms |

Dễ dàng nhận thấy thời gian của việc sử dụng những hàm có sẵn nhanh hơn so với chương trình chúng ta tự cài đặt.

Tuy nhiên xét về độ chính xác, các thuật toán tự cài đăt đạt được độ chính xác cao, rất khớp với việc sử dụng hàm do OpenGL cung cấp.

1. Hướng dẫn sử dụng

Chương trình đọc dữ liệu đầu vào thông qua file **input.txt**, mỗi dòng trong file sẽ chứ các tham số cho từng hình (mã hình và các giá trị điều khiển), đảm bảo các dòng phải đủ các giá trị để chương trình có thể thực hiện đúng

File thực thi phải nằm cùng thư mục với file **input.txt** để có thể đọc dữ liệu, chỉ cần chạy file thực thi, sẽ xuất hiện màn hình GLUT chứa các hình vẽ, màn hình Console hiển thị thông tin hình vẽ và thời gian vẽ hình đó.

Để hiển thị hình vẽ và thời gian chạy các thuật toán dựa vào hàm do OpenGL cung cấp, mở comment trong file Draw2DObject.cpp để chạy.

1. Tài liệu tham khảo
2. Các bài giảng của thầy Trần Thái Sơn và hướng dẫn thực hiện đồ án của thầy Võ Hoài Việt
3. [How to setup OpenGL with Visual Studio 2019 on Windows 10?](https://www.geeksforgeeks.org/how-to-setup-opengl-with-visual-studio-2019-on-windows-10/)
4. [Thuật toán vẽ đường thẳng DDA](https://tuhoclaptrinh.cachhoc.net/2017/02/25/bai-3-thuat-toan-ve-duong-thang-dda/)
5. [Thông tin về hình Parabol](https://vi.wikipedia.org/wiki/Parabol)
6. [Thông tin về hình Hyperbol](https://vi.wikipedia.org/wiki/Hyperbol)