**Bài tập 2:**

Phần 1:

1. Nét đứt:

- Thuật toán sử dụng: DDA.

- Sử dụng thuật toán DDA xong lấy ra list các điểm được sinh ra sau đó put pixel cho mỗi 5 điểm 1 lần(for….; …; i+=5). Tạo biến lưu thông tin số lần put pixel ( ở đây nhóm sử dụng 5 lần put pixel thì mới có 1 khoảng trắng và khoảng trắng tương đương với 3 lần put pixel) nếu số lần put chia hết cho 5 thì ta tiến hành dừng put pixel và bắt đầu đếm số lần không put pixel. Nếu số lần không put pixel bằng 3 thì reset số lần put pixel về không là tiếp tục tiền hành đến khi hết list.

1. Nét chấm gạch:

- Thuật tóa sử dụng : DDA.

- Tương tự nét đứt, ta thêm một biến đếm số lần put dấu chấm (int countDot ). Giống với nét đứt, sau khi ta tạo ra một khoảng trắng thì bắt đầu put pixel 1 lần rồi cộng số lần tạo khoảng trắng thêm 1. Sau khi số lần tạo khoảng trắng = 2 và số lần put dot =1 thì ta reset số lần put pixcel của đường thẳng rồi tiếp tục thực hiện vòng lặp.

1. Nét 2 chấm gạch:

-Thuật toán sử dụng: DDA.

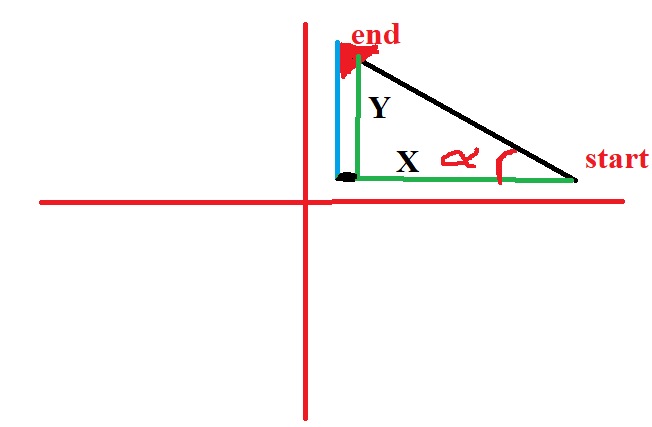
-Tương tự nét chấm gạch, lần này ta cần 2 lần put dot và 3 lần khoảng trắng rồi mới reset và tiếp tục put pixcel 5 lần.

1. Đường thẳng có dấu mũi tên:

-Thuật toán sử dụng: DDA.

-Vẽ đường thẳng bằng các cách trên và sau đó tiến hành vẽ dấu mũi tên.

-Giả sử ta có đường thẳng:



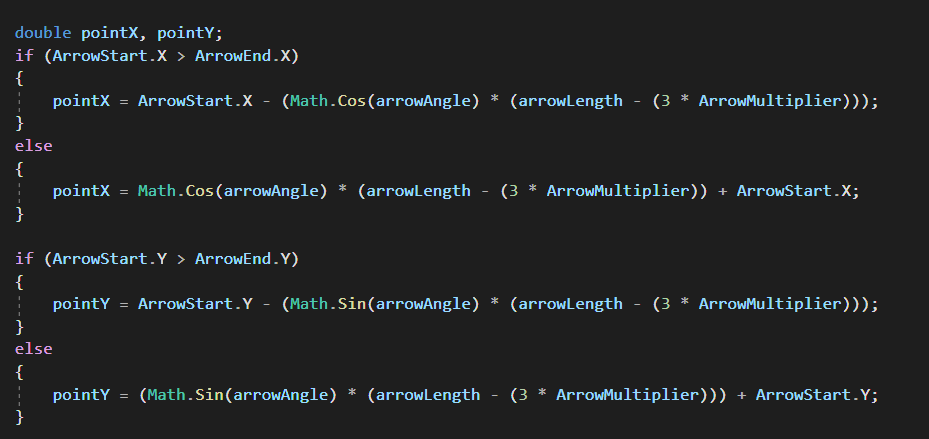
Ta tính độ dài của đoạn thằng = công thức :



Sau đó tìm ra góc của dấu mũi tên ( góc alpha trên hình minh họa) bằng cách:



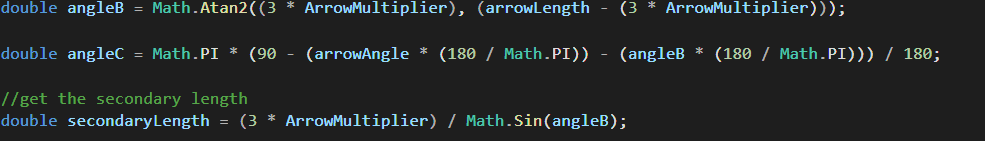
Tiếp đến ta tìm ra độ dài của X, Y (có tổ hợp 4 trường hợp):



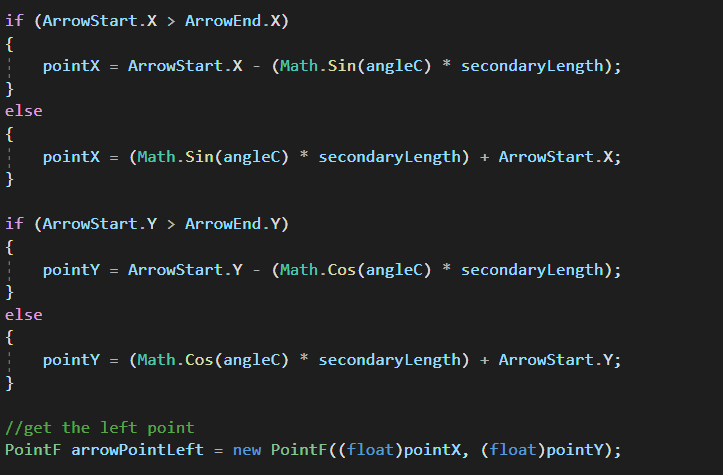
Sau khi lấy được X và Y thì ta có được tọa độ của điểm cuối của dấu mũi tên trên đường thẳng:



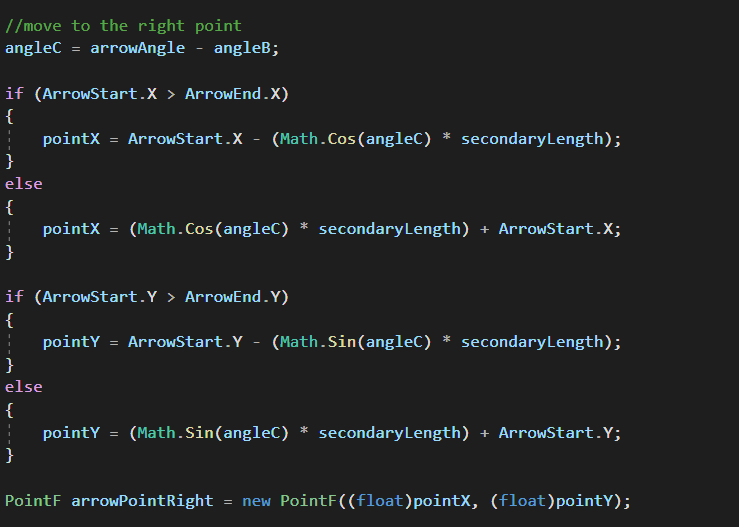
Bước tiếp theo là tìm ra góc từ đỉnh nối tới điểm bên trái của mũi tên so với trục và góc từ điểm bên trái của mũi tên nối vuông góc tới đường thẳng so với trục và độ dài còn lại của đoạn thẳng sau khi cắt đi phần chứa mũi tên:



Sau đó ta cũng tính được điểm bên trái của mũi tên tương tự như trên:



Và tương tự ta có được điểm bên phải của mũi tên :



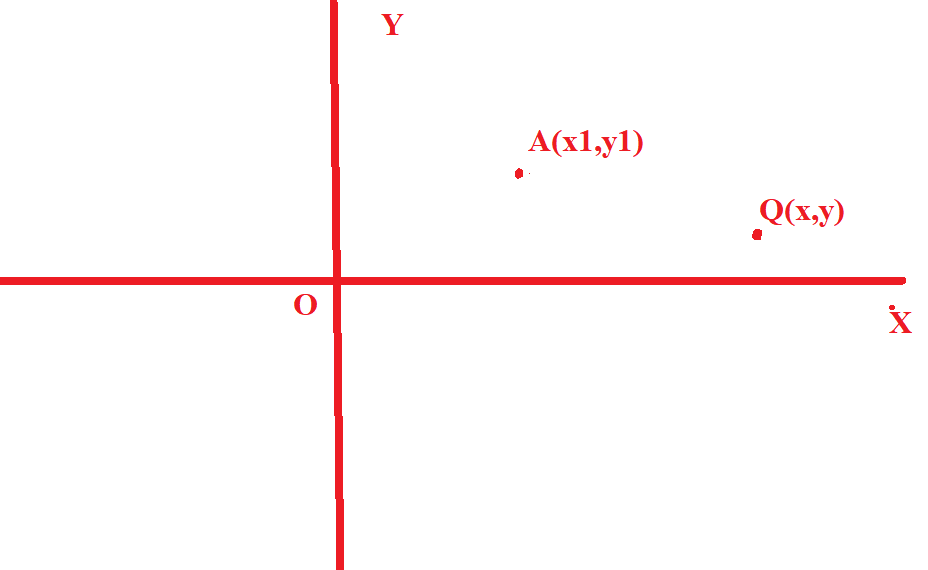
Sau khi có được điểm bên trái và điểm bên phải thì ta dùng thuạt toán DDA để vẽ các đường thẳng từ đầu tới trái, đầu tới phải, trái tới phải.

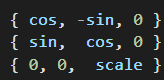
Phần 2:

-Thuật toán vẽ đường tròn nét đứt tương tụ như đường thẳng nét đứt. Tạo biến lưu thông tin số lần put pixel ( ở đây nhóm sử dụng 5 lần put pixel thì mới có 1 khoảng trắng và khoảng trắng tương đương với 3 lần put pixel) nếu số lần put chia hết cho 5 thì ta tiến hành dừng put pixel và bắt đầu đếm số lần không put pixel. Nếu số lần không put pixel bằng 3 thì reset số lần put pixel về không là tiếp tục tiền hành đến khi hết list.

**Bài tập 5:**

1)



-Để tìm được điểm P(x2,y2) sau khi quay Q quanh A, bước đầu tiên ta cần tịnh tiến điểm Q theo A ( x-x1, y-y1) về góc tọa độ. Sau đó ta thực hiện thao tác nhân điểm Q với ma trận .

- Sau khi nhân Q với ma trận biến đổi ta được Q(x1s, y1s) sau đó ta tịnh tiến Q ngược về (x3+x1, y3+y1) thì ta đã tìm được P(x2,y2).

2)

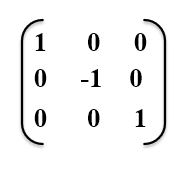
Để tìm điểm P(x3,y3) đối xứng với Q(x,y) qua đường thẳng A(x1,y1) B(x2,y2) ta cần:

-B1: Tịnh tiến cả 3 điểm về góc tọa độ O thông qua A(x1, y1).

=> Q(x-x1,y-y1), A(0,0) B(x2-x1,y2-y1).

-B2: Tiến hành quay AB quanh O sao cho AB trùng với Ox ta lấy được góc quay Alpha.

-B3: Quay Q quanh O với góc alpha ta được điểm Q1.

-B4: Tìm điểm P1 là điểm đối xứng với Q1 qua Ox bằng cách nhân Q1 với ma trận biến đổi .

-B5: Sau khi tìm được điểm P1 ta tiến hành quay trả P1 về 1 góc -Alpha rồi cộng P1 với (x1,y1) đã trừ lúc đầu ta được điểm P(x3,y3) là điểm đối xứng với Q qua AB.

**Bài tập 6:**

A

Q

**Y**

**X**

**Z**

Ta có:

**O**

* **Tịnh tiến A(x1,y1,z1) sao cho A trùng góc tọa độ và điểm Q’(x’,y’,z’) là điểm tịnh tiến của Q(x,y,z) theo A:**

A(x1,y1,z1) -> O(0,0,0) => A’ = A\*

Với =

Q(x,y,z) -> Q’(x’, y’, z’) với Q’ = Q \*

=> (x’, y’, z’, 1) = (x, y, z, 1) \* = (x – x1, y – y1, z – z1, 1)

=> Q’(x – x1, y – y1, z – z1)

A

Q

**Y**

**X**

**Z**

A'

Q’

**O**

* **Lấy P’(x2’,y2’,z2’) là đối xứng của Q’(x – x1, y – y1, z – z1) qua O:**

P’ = Q’ \* với

=> (x2’, y2’, z2’, 1) = (x – x1, y – y1, z – z1, 1) \*

= (– x+x1, – y +y1, – z + z1, 1)

=> P’ (– x+x1, – y +y1, – z + z1)

P’

Q’

A’

**Y**

**X**

**Z**

**O**

* **Tịnh tiến A’ trở về vị trí cũ, P’ tịnh tiến theo A’ ta được điểm P(x2,y2,z2) cần tìm:**

A’(0,0,0) -> A(x1,y1,z1) ⬄ A = A’ \*

với =

P’ (– x+x1, – y +y1, – z + z1) -> P(x2,y2,z2)

=> P = P’ \*

=> (x2, y2, z2, 1) = (– x+x1, – y +y1, – z + z1) \*

= (2x1 – x, 2y1 – y, 2z1 – z, 1)

=> **P(2x1 – x, 2y1 – y, 2z1 – z)**

P

P’

Q’

**O**

A

Q

**Y**

**X**

**Z**

***Kết luận:***

Ma trận biến đổi của điểm P(x2,y2,z2) là đối xứng của điểm Q(x,y,z) qua điểm A(x1,y1,z1) bất kì là:

**P(x2, y2, z2) = Q(x, y, z) \* \* \***