**Aluno: André Luiz N. Carneiro De Castro RA: 92854**

##### P1 - Computação Gráfica



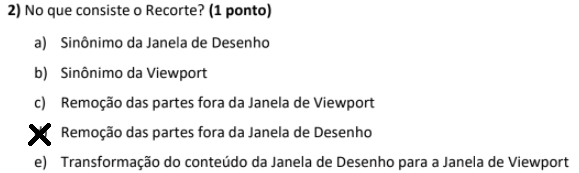
Primeira Avaliação.

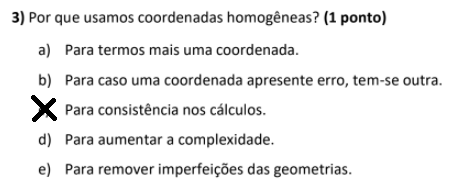
**Prof. M.e Tiago Remedio**

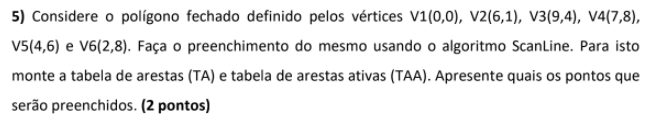
#### ARARAS/SP

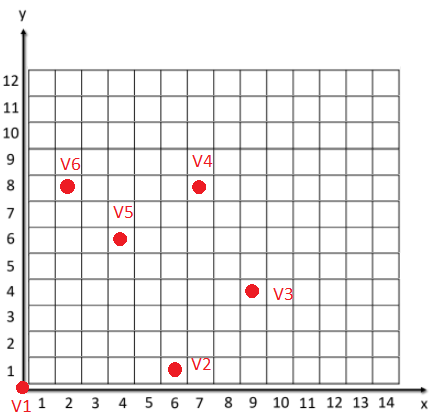
#### 10/2020

#### 









Preenchendo os vértices temos:

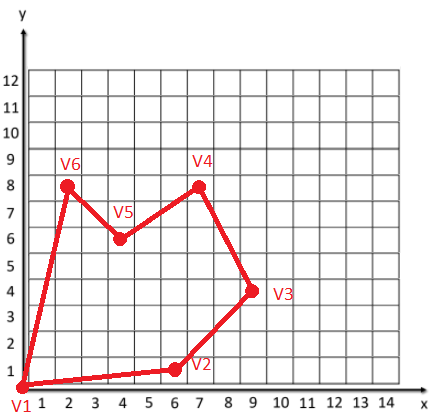


Figura 1 - Polígono obtido através dos vértices

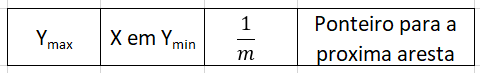
Funcionamento do Algoritmo que funcionará como um scanner:

1° Passo do algoritmo é ter todas as arestas modeladas.

2° Tabela de arestas (organizam quais arestas serão analisadas a cada vez que a linha passar).

3° Tabela de arestas ativas, linha de *scanline* bateu e terá que pintar.

#### Começará lendo as linhas horizontais. E sempre iremos desconsiderar os vértices superiores.



Estrutura da Aresta

Partindo como início pelo Y, seguindo a sequência de AB, primeira aresta, temos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V12 | 1 | 0 | 6 | 0(ymin) |

= =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V23 | 4 | 6 | 1 | 1(ymin) |

= =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V34 | 8 | 9 | -2/4 | 4(ymin) |

= =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V45 | 8 | 4 | -3/-2 | 6(ymin) |

= =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V56 | 8 | 4 | -1 | 6(ymin) |

= =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V61 | 8 | 0 | -2/-8 | 0(ymin) |

= =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V12 | 1 | 0 | 6 | 0(ymin) |
| V23 | 4 | 6 | 1 | 1(ymin) |
| V34 | 8 | 9 | -0,5 | 4(ymin) |
| V45 | 8 | 4 | 1,5 | 6(ymin) |
| V56 | 8 | 4 | -1 | 6(ymin) |
| V61 | 8 | 0 | 0,25 | 0(ymin) |

Organizando tabela de arestas, que serão organizados através do Y mínimo de cada um, teremos:

Y atual: 0

Entrará [ 8 , 0 , 0.25 , 0 ]

Entrará [ 1 , 0 , 6.0 , 0 ]

Tabela de Arestas Ativas (TAA) Ordenada em x:

Pos 0 [ 8 , 0 , 0.25 , 0 ]

Pos 1 [ 1 , 0 , 6.0 , 0 ]

Pintará de ( 0 , 0 ) até ( 0 , 0 )

Y atual: 1

Entrará [ 4 , 6 , 1.0 , 1 ]

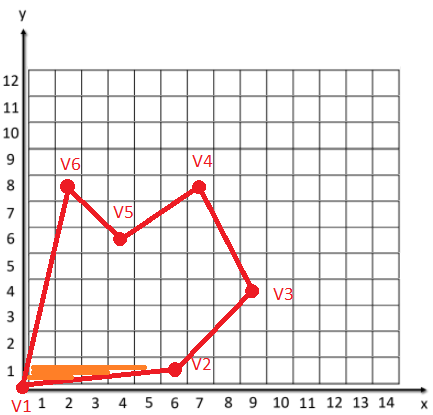
Sairá [ 1 , 6.0 , 6.0 , 0 ]

TAA - Ordenada em X

0 [ 8 , 0.25 , 0.25 , 0 ]

1 [ 4 , 6 , 1.0 , 1 ]

Pintará ( 0 , 1 ) até ( 6 , 1 )



Y atual: 2

TAA Ordenada em x!

0 [ 8 , 0.5 , 0.25 , 0 ]

1 [ 4 , 7.0 , 1.0 , 1 ]

Pintará ( 1 , 2 ) até ( 7 , 2 )

Y atual: 3

TAA Ordenada em x!

0 [ 8 , 0.75 , 0.25 , 0 ]

1 [ 4 , 8.0 , 1.0 , 1 ]

Pintará ( 1 , 3 ) até ( 8 , 3 )

Y atual: 4

Entrou [ 8 , 9 , -0.5 , 4 ]

Sairá: [ 4 , 9.0 , 1.0 , 1 ]

TAA - Ordenada em x!

0 [ 8 , 1.0 , 0.25 , 0 ]

1 [ 8 , 9 , -0.5 , 4 ]

Pintará ( 1 , 4 ) até ( 9 , 4 )

Y atual: 5

TAA - Ordenada em x!

0 [ 8 , 1.25 , 0.25 , 0 ]

1 [ 8 , 8.5 , -0.5 , 4 ]

Pintará ( 1 , 5 ) até ( 8 , 5 )

Y atual: 6

Entrou [ 8 , 4 , -1.0 , 6 ]

Entrou [ 8 , 4 , 1.5 , 6 ]

TAA - Ordenada em x!

0 [ 8 , 1.5 , 0.25 , 0 ]

1 [ 8 , 4 , -1.0 , 6 ]

2 [ 8 , 4 , 1.5 , 6 ]

3 [ 8 , 8.0 , -0.5 , 4 ]

Pintará ( 2 , 6 ) até ( 4 , 6 )

Pintará ( 4 , 6 ) até ( 8 , 6 )

Y atual: 7

TAA - Ordenada em x!

0 [ 8 , 1.75 , 0.25 , 0 ]

1 [ 8 , 3.0 , -1.0 , 6 ]

2 [ 8 , 5.5 , 1.5 , 6 ]

3 [ 8 , 7.5 , -0.5 , 4 ]

Pintará ( 2 , 7 ) até ( 3 , 7 )

Pintará ( 6 , 7 ) até ( 7 , 7 )

Y atual: 8

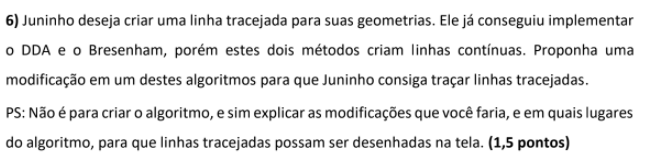
Sairá [ 8 , 7.0 , -0.5 , 4 ]

Sairá [ 8 , 7.0 , 1.5 , 6 ]

Sairá [ 8 , 2.0 , -1.0 , 6 ]

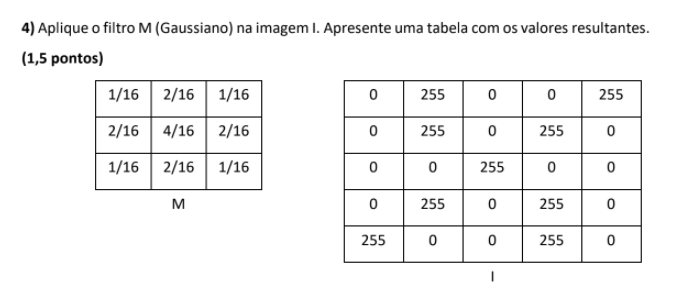
Sairá [ 8 , 2.0 , 0.25 , 0 ]

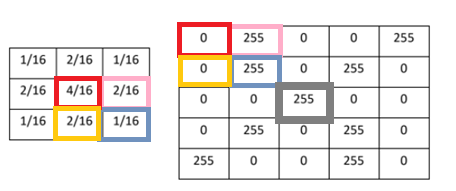
Com isso teremos o fim do Scanline e nosso polígono preenchido.



Juninho deveria modificar o Bresenham para que ele tenha a função de um Bresenham Baixa/Alta para realizar as linhas tracejadas nas *diagonais*.

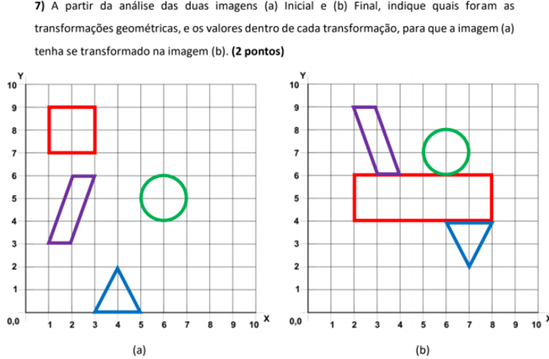
Para as linhas *verticais* e *horizontais*, ele poderia utilizar DDA com uma pequena alteração adicionando um contador. Como o DDA pinta pixels que estão próximos, essa mecânica de distanciamento dos pixels faz com que as linhas fiquem tracejadas. Também funcionaria para linhas diagonais.





Analisando os pixels centrais da tabela e multiplicando os valores correspondentes somamos e atualizamos os pixels, obtemos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 47,8 | 95,6 | 63,7 | 63,7 | 79,6 |
| 47,8 | 111,6 | 111,6 | 95,6 | 63 |
| 31 | 95,6 | 127,5 | 95,6 | 31,9 |
| 63,7 | 95,6 | 111,6 | 111,6 | 47,8 |
| 79,6 | 63,7 | 63,7 | 95,6 | 47,8 |



1. Primeiramente *TODOS* os polígonos devem passar pela translação até o ponto de origem, para que elas tenham um ponto de referência.
2. Polígono roxo, espelhamento em y, translocado 3 unidades em y com valor positivo e 1 unidade em X positiva.
3. Polígono laranja, translocado em x em 1 unidade e translocado em Y negativamente em 3 unidades. Após isso sofreu uma escala em X de 3 unidades positivas.
4. Polígono azul, espelhamento em x, translação em x de 3 unidades para a direita e translocado em y com valor de 4 unidades.

1. Polígono verde translocado em Y em 2 unidades com valor positivo.