QUESTÃO 1 (3 valores) (1.1 = 1.0; 1.2 =2.0)

Recorra ao algoritmo *scan-line* para calcular as coordenadas dos *pixels* de preenchimento da área bidimensional definida pelo polígono constituído pelos vértices $\boldsymbol{A}(4,0)$, $\boldsymbol{B}(6,2)$ *e* $\boldsymbol{C}(2,2)$

1.1. Apresente o estado da tabela de arestas (ET - *Edge Table*) e tabela de arestas activas (AET - *Active Edge Table*) no início do algoritmo.

1.2. Calcule as coordenadas dos pixels de preenchimento até à 3ª iteração, apresentando cada iteração do algoritmo separadamente, indicando o estado da ET e AET, e apresente no final, de forma gráfica, o preenchimento.

R:

1.1.)

Antes de iniciar as iterações do algoritmo é necessário obter para cada aresta do polígono que vai ser preenchido, um conjunto de elementos informativos sobre os mesmos, tais como: declive m e $\frac{1}{m}$, $x_{mínimo}$, $y_{mínimo}$ e $y_{máximo}$ por forma a podermos construir as tabelas ET e AET.

Nota: $x_{mínimo}$ é a coordenada que acompanha $y_{mínimo}$.

Para cada aresta do polígono, vista como segmento de reta, obtemos os valores de $m = \frac{dy}{dx}$ tendo em conta que cada aresta $\overline{A_1A_2}$ é obtida através dos pontos sucessivos do polígono ligando o último ponto ao primeiro.

Temos assim:

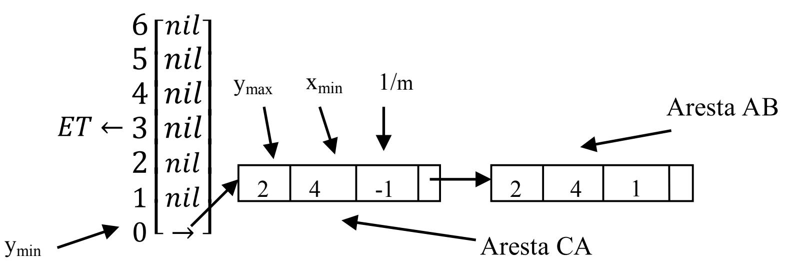
$$\overline{AB} = [(4,0),(6,2)]: m = \frac{2-0}{6-4} = \frac{2}{2}; \frac{1}{m} = 1; y_{min} = 0; x_{min} = 4;\ y_{máx} = 2;$$

$$\overline{BC} = [(6,2),(2,2)]: m = \frac{2-2}{2-6} = \frac{0}{-4} = 0; \frac{1}{m} = \infty; \text{aresta horizontal, logo é ignorada}$$

$$\overline{CA} = [(2,2),(4,0)]: m = \frac{0-2}{4-2} = \frac{-2}{2} = -1; \frac{1}{m} = -1; y_{min} = 0; x_{min} = 4;\ y_{máx} = 2;$$

No arranque do algoritmo as tabelas ET e AET tem o seguinte estado:

$AET \leftarrow nil$



2.2)

Seja a linha de varrimento (*scan-line*) definida por *SL* sendo que o varrimento se faz na vertical, percorrendo o eixo Y desde os valores mais pequenos até à maior coordenada em Y do polígono.

$SL \leftarrow 0$ ou seja, a linha de varrimento recebe o valor do primeiro índice da entrada da tabela ET que não esteja vazia, neste caso a entrada 0, que corresponde à coordenada *y* menor do polígono partilhada pelas arestas $\overline{AB}$ e $\overline{CA}$.

Iteração do algoritmo

Até que *AET* fique vazia repete

<u>Iteração 1</u>

1.1) Mover de *ET* para *AET* as arestas que passam a ser intersectadas por *SL*, ou seja, o valor de entrada com índice *SL*.
A tabela *AET* recebe então os valores da(s) aresta(s) de *ET* com entrada *SL*.

$AET \leftarrow \overbrace{|2|4|-1|}^{\overline{CA}} \rightarrow \overbrace{|2|4|1|}^{\overline{AB}}$

1.2) Eliminar de *AET* todas as arestas que deixam de ser intersectadas pela linha de varrimento *SL*. Basta verificar se existe alguma aresta em *AET* cujo $y_{máximo} \leq SL$

Neste caso temos *SL* = 0 e $\overline{CA}\ y_{máximo} = 2$ e $\overline{AB}\ y_{máximo} = 2$ logo $SL <$ que ambos. Não existem arestas em condições de ser eliminadas de *AET*, ficando esta tabela inalterada.

1.3) Identificar pontos a desenhar no ecrã entre pares de arestas de *AET* (estas são as intersectadas por *SL*).
A coordenada em *y* é o valor de *SL* e temos de calcular os valores da abcissa *x* mais à esquerda e mais à direita (daqui a importância do valor de $x_{mínimo}$ a abcissa que acompanha a ordenada do ponto de intersecção de SL com cada aresta).

Ou seja, vamos considerar os pontos entre $x_{mínimo}$ de pares de arestas em *AET*.
Neste caso, $\overline{CA}\ x_{mínimo} = 4$ e $\overline{AB}\ x_{mínimo} = 4$ logo
Pontos a ativar: $(4,1)$
SL

1.4) Incrementamos SL ou seja $SL \leftarrow SL + 1; SL = 1$

1.5) Atualizamos AET de $x_{mínimo}$ tendo em conta o incremento *SL* vem

$x_{i+1} \leftarrow x_i + \frac{1}{m}$ para qualquer aresta.

Temos então:

$\overline{CA}$: $x_{i+1} \leftarrow 4 + (-1) = 3$; $\overline{AB}$: $x_{i+1} = 4 + 1 = 5$

Ficando $AET \leftarrow \overbrace{|2|3|-1|}^{\overline{CA}'} \rightarrow \overbrace{|2|5|1|}^{\overline{AB}'}$

<u>Iteração 2</u>

2.1) Existem arestas na entrada *SL (==1)* de *ET* ? Não.

2.2) Existem arestas em AET tal que $y_{máximos} \leq SL$? Não.

2.3) Pontos a ativar no ecrã entre pares de arestas de AET.
Abcissas entre $x = 3\ (\overline{CA}'\ x_{mínimo})$ e $x = 5\ (\overline{AB}'\ x_{mínimo})$

Pontos a ativar: $(3,1), (4,1), (5,1)$ *(recordar que SL = 1, ordenado y dos pontos a preenche)*

2.4) Incrementamos SL ou seja $SL \leftarrow SL + 1; SL = 2$
2.5) Atualizamos AET tal que
$\overline{CA}': x_{i+1} \leftarrow\ 3 + (-1) = 2;\ \overline{AB}': x_{i+1} = 5 + 1 = 6$

Ficando $AET \leftarrow \overbrace{|2|2|-1|}^{\overline{CA}''} \rightarrow \overbrace{|2|6|1|}^{\overline{AB}''}$

Iteração 3

3.1) Existem arestas na entrada *SL (==2)* de *ET* ? Não.

3.2) Existem arestas em AET tal que $y_{máximos} \leq SL$? Sim.

Nota: retirando as arestas CA´´ e AB´´ da *AET* implica que os pontos de fronteira não são ativados, e o algoritmo termina aqui.

Se se pretender ativar os pontos de fronteira (pontos da aresta BC que foi ignorada por ser horizontal), basta que o teste no segundo passo seja pela desigualdade apenas.

Vamos aplicar esta alteração:

Existem arestas em AET tal que $y_{máximos} < SL$? Não.

3.3) Pontos a ativar no ecrã entre pares de arestas de AET.
Abcissas entre $x = 2\ (\overline{CA}''\ x_{mínimo})$ e $x = 6\ (\overline{AB}''\ x_{mínimo})$

Pontos a ativar: $(2,2), (3,2), (4,2), (5,2), (6,2)$

3.4) Incrementamos SL ou seja $SL \leftarrow SL + 1; SL = 3$
3.5) Atualizamos AET tal que
$\overline{CA}'': x_{i+1} \leftarrow\ 2 + (-1) = 1;\ \overline{AB}'': x_{i+1} = 6 + 1 = 7$

Ficando $AET \leftarrow \overbrace{|2|1|-1|}^{\overline{EA}'''} \rightarrow \overbrace{|2|7|1|}^{\overline{AB}'''}$

Representação gráfica do resultado final, com pontos ativados.