

# COMPUTAÇÃO GRÁFICA E INTERFACES

FCT/UNL – Ano letivo 2016/2017

Teste 2A – 2016.12.12

Responda no próprio enunciado, que entregará. Em caso de engano, e se o espaço para as respostas não for suficiente, poderá usar o verso das folhas desde que feitas as devidas referências.

**Não desagrafe as folhas! A prova, com duração de 2H, é sem consulta!**

## 1. (3.5 valores)

Assinale com V (Verdadeiro) ou F (Falso) as afirmações abaixo. Cada resposta errada desconta 50% da sua cotação.

Na projeção axonométrica, o paralelismo das linhas é preservado.	
Na projeção perspetiva, o paralelismo das linhas não é preservado em caso algum.	
Numa projeção axonométrica com ambos os parâmetros não nulos, os ângulos nunca são preservados.	
Na projeção perspetiva o número de pontos de fuga está relacionado com a posição do centro de projeção.	
A projeção que melhor se adequa ao foto-realismo é a projeção axonométrica.	
A função <code>lookAt()</code> devolve uma matriz que inclui uma translação e uma rotação.	
A função <code>lookAt()</code> permite especificar uma matriz de projeção axonométrica arbitrária.	
A matriz devolvida pela função <code>lookAt()</code> permite converter pontos de coordenadas da câmara para coordenadas do mundo.	
Todos os problemas de <i>aliasing</i> podem ser minorados por um aumento da resolução do dispositivo.	
O <i>antialiasing</i> por filtragem posterior permite aumentar a nitidez da imagem, aumentando o contraste.	
O algoritmo de recorte de Sutherland-Hodgeman pode ser generalizado para janelas não retangulares, desde que convexas.	
Num grafo de cena, se trocarmos a ordem dos ramos do grafo, a cena produzida não é alterada.	
Num grafo de cena otimizado, podemos dispensar o uso da pilha de transformações e manter apenas a matriz de transformação corrente.	
Num grafo de cena, a ordem das transformações encontradas num mesmo ramo é irrelevante para o resultado final, pois elas serão sempre efetuadas pela ordem T.R.S	

## 2. (2.5 valores)

Considere o modelo de reflexão ambiente e reflexão difusa estudado nas aulas,  $I_{rgb} = I_a K_a + I_p K_d \cos \theta$ , avaliado num ponto arbitrário da superfície dum objeto. Assinale com V (Verdadeiro) ou F (Falso) as afirmações abaixo. Cada resposta errada desconta 50% da sua cotação.

A luz refletida na direção do observador depende da orientação da superfície em relação ao observador.	
A luz refletida na direção do observador depende da orientação da superfície em relação à fonte de luz.	
$I_a$ é um valor escalar.	
$I_p$ é um vetor.	
$I_p$ está associado a uma fonte de luz particular.	
$I_a$ está associado a uma fonte de luz particular.	
$K_d$ representa a cor do objeto, quando iluminado por uma luz branca, para a reflexão difusa.	
Para várias fontes de luz repete-se o segundo termo, mas mudando o valor de $K_d$ .	
A reflexão difusa é máxima quando a luz incide perpendicularmente à superfície.	

## 3. (3 valores)

Considere que numa aplicação 3D pretende oferecer ao utilizador a possibilidade de especificar um volume de visão e tipo de projeção o mais semelhante possível aos obtidos por uma câmara fotográfica comum.

a) De entre as funções estudadas nas aulas para a definição do volume de visão:

1. `frustum(left, right, bottom, top, near, far)`
2. `perspective(fovy, aspect, near, far)`
3. `ortho(left, right, bottom, top, near, far)`,

b) Como sabe, qualquer das funções referidas em a) produzem uma matriz capaz de transformar o volume de visão em causa num volume dito canónico, pré-definido, e que se estende de -1 a 1 em cada um dos 3 eixos. Qual a importância dessa transformação na implementação do pipeline gráfico 3D?

c) A função  $\text{lookAt}(\text{eye}, \text{at}, \text{up})$  estudada nas aulas permite estabelecer uma relação entre o referencial da câmara e o referencial do mundo (WC). Considere que a matriz devolvida por aquela função é designada aqui por  $\mathbf{M}$ . Preencha os espaços em branco para tornar válidas as igualdades abaixo.

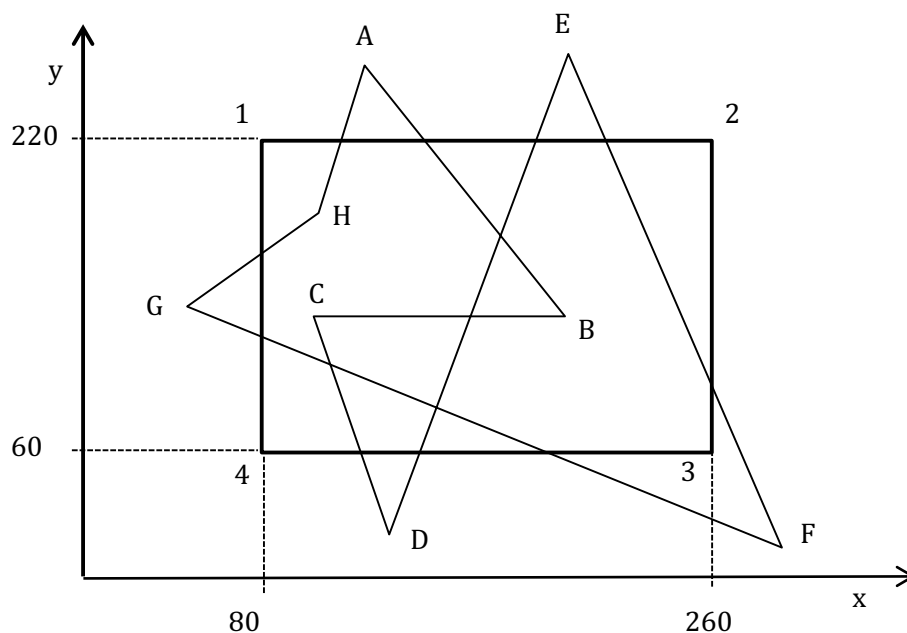
$$\mathbf{M} \cdot \text{eye} = \underline{\hspace{2cm}}$$

$$\mathbf{M} \cdot \underline{\hspace{2cm}} = [0 \ 1 \ 0]^T$$

$$(1/||\text{at-eye}||) \mathbf{M}(\text{at-eye}) = \underline{\hspace{2cm}}$$

#### 4. (5 valores)

São dados os polígonos  $P=[A, B, C, D, E, F, G, H]$  e  $Q=[1, 2, 3, 4]$ , sendo este último considerado a janela de recorte, com as respectivas coordenadas indicadas na figura.



O polígono P vai ser recortado pela janela Q aplicando o algoritmo de Sutherland-Hodgeman, para o qual se estipulou a seguinte ordem: Clip Left -> Clip Bottom -> Clip Right -> Clip Top.

a) Indique os polígonos resultantes de cada uma das fases de recorte:

Clip Left:

Clip Bottom:

Clip Right:

Clip Top:

b) Quantas arestas possui o polígono final recortado? \_\_\_\_\_

Imagine agora que, em vez de se recortar o polígono P, seriam as suas arestas a ser recortadas pela janela Q usando o algoritmo de Cohen-Sutherland.

c) Preencha a tabela seguinte com a decisão do algoritmo para os segmentos de reta indicados (**aceitação/rejeição** trivial ou **interseção** com reta limite da janela), bem como o número total de interseções de cada segmento com as retas limites da janela de recorte e a indicação da primeira interseção com essas mesmas retas limite efetivamente determinada pelo algoritmo. Admita que os bits são atribuídos da esquerda para a direita, usando a mesma ordem que a definida na alínea a), ou seja, Left, Bottom, Right, Top, sendo essa mesma ordem a da progressão do algoritmo.

d)

Aresta	Decisão	Número <b>total de interseções</b> com as retas limite da janela de recorte	Equação da reta correspondente à determinação efetiva da <b>1ª interseção</b>
AB			
BC			
CD			
DE			
EF			
FG			
GH			
HA			

e) Pinte, na figura, a área que seria preenchida por aplicação do algoritmo FILL AREA ao polígono **inicial** P!

f) Quantas entradas não vazias teria a tabela de arestas (TA)? Justifique indicando os índices de cada uma dessas entradas, bem como as arestas nelas constantes. **Nota:** para identificar uma entrada poderá indicar o seu índice recorrendo à ordenada dum ponto (p.ex.  $y_B$ ):

g) Apresente o conteúdo da Tabela de Arestas Ativas (TAA), imediatamente antes do preenchimento da linha de varrimento corrente, para cada uma das seguintes linhas de varrimento:

$y_3$ :

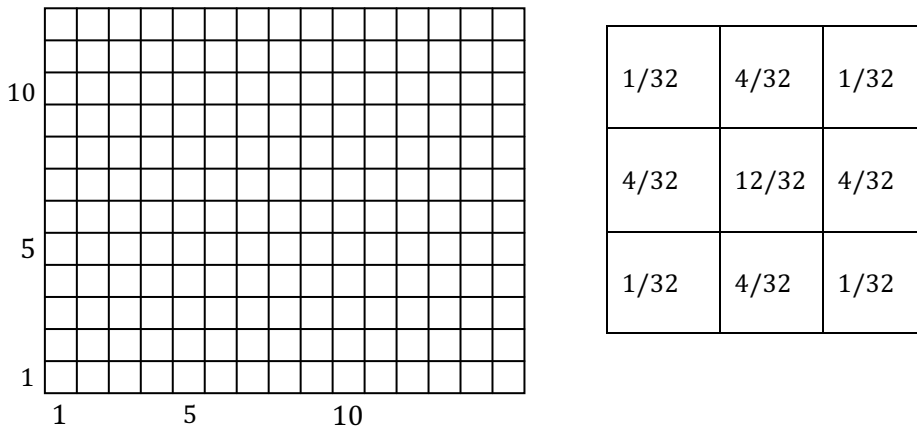
$y_D$ :

$y_H$ :

$y_C$ :

### 5. (3 valores)

Nas duas figuras seguintes, cada quadrícula representa um pixel num sistema gráfico em tons de cinzento. A cor branca, do fundo, é representada pelo valor 255, enquanto a cor preta, a qual será usada para desenhar as primitivas, é representada pelo valor 0.



- a) Pinte, na figura do lado esquerdo, as quadrículas correspondentes aos pixels que resultariam da aplicação do algoritmo do Ponto Médio ao segmento de reta cujas extremidades são os pixels (2,2) e (13,5).
- b) Ao resultado obtido em a) aplicou-se um filtro (apresentado no lado direito da mesma figura) para suavizar os efeitos provocados pelo *aliasing*. Qual a cor final do pixel de coordenadas (3,3) após a aplicação do referido filtro? Apresente a sua solução indicando os cálculos necessários, mas sem os efetuar.
- c) Veria algum inconveniente na aplicação do método referido em b) para suavizar texto? Justifique a sua resposta!
- d) Explique, com detalhe, como aplicaria o filtro nas margens da imagem.

### 6. (3.0 valores)

O seguinte programa WebGL já se encontra otimizado em termos de operações Push e Pop.

- |                            |                           |
|----------------------------|---------------------------|
| 1. multTranslate([10,0,0]) | 14. P2()                  |
| 2. multScale([2,2,1])      | 15. PopMatrix()           |
| 3. PushMatrix()            | 16. multRotY(35°)         |
| 4. multRotY(40°)           | 17. multScale([3,2,2])    |
| 5. multScale ([2,1,2])     | 18. multRotX(25°)         |
| 6. PushMatrix()            | 19. PushMatrix()          |
| 7. multTranslate ([0,1,0]) | 20. multRotZ(10°)         |
| 8. multRotZ(5°)            | 21. P3()                  |
| 9. multRotX(30°)           | 22. PopMatrix()           |
| 10. P1()                   | 23. multScale([3,1,0.5])  |
| 11. PopMatrix ()           | 24. multTranslate(2,0,0]) |
| 12. multRotX(-30°)         | 25. P4()                  |
| 13. multRotZ(20°)          |                           |

- a) Desenhe o grafo de cena correspondente ao programa apresentado, sem proceder a qualquer possível simplificação:

- b) Se, num determinado sistema apenas fosse permitida a criação de nós de transformação compostos, com a seguinte ordem fixa de transformações elementares T.Rz.Ry.Rx.S, aqui designados por *super-nós*. Qual seria o número mínimo de nós que seriam necessários para desenhar a cena correspondente ao código apresentado?

Assinale no grafo os ramos que, pela disponibilização dos *super-nós*, veriam o número de nós usados ser reduzido.