

COMPUTAÇÃO GRÁFICA E INTERFACES

Ano letivo 2016/2017 – Teste 1 – 2016.11.03

Atenção

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 **1H30**, é **sem consulta**!

1. (3 valores)

Assinale com V (Verdadeiro) ou F (Falso) as afirmações abaixo. Cada resposta errada desconta 50% da sua cotação. No pipeline de visualização do WebGL:

O <i>fragment shader</i> pode acabar por não ser executado quando se desenha, por exemplo, um triângulo.	
O CPU (programa javascript) envia pixels para o pipeline, sendo o GPU responsável por pintá-los através da execução dum <i>fragment shader</i> .	
Podemos associar valores (cores, vetores normal, etc.) às primitivas gráficas que são produzidas pela aplicação.	
Um <i>fragment shader</i> pode receber dados arbitrários que não os produzidos ao nível de cada vértice pelo <i>vertex shader</i> .	
Uma aplicação representa sempre as coordenadas dos vértices usando o tipo de dados vec4 .	
Podemos mudar de programa GLSL (<i>vertex+fragment shader</i>) a meio duma primitiva.	
As variáveis de tipo uniform correspondem a valores que a aplicação javascript envia exclusivamente ao <i>vertex shader</i> .	
As variáveis de tipo varying contêm valores conhecidos pela aplicação javascript.	
O resultado dum <i>vertex shader</i> é completamente arbitrário, não existindo qualquer obrigatoriedade na produção de um valor concreto.	
O resultado dum <i>fragment shader</i> terá sempre que conter a cor do pixel, no caso do fragmento não ser descartado.	

2. (3 valores)

Os seguintes *shaders* fazem parte dum mesmo programa GLSL.

a) complete o código apresentado preenchendo os espaços:

```
uniform ____ m;
____ vPos;
____ vfoo;
varying vec4 ffoo;
void main() {
    ____ = m * vPos;
    ____ = vec4(vfoo, 1.0);
}

____ ffoo;

void main() {
    ____ = ffoo;
}
```

b) indique uma possível utilização para o identificador `vfoo`:

3. (4 valores)

Um ecrã dum dispositivo móvel *tablet* possui uma resolução de 1280x800 pixels. A metade esquerda do ecrã será usada para apresentar texto enquanto que a metade direita do ecrã está ainda dividida em duas áreas de igual dimensão, uma por cima da outra. Uma janela 2D, definida em World Coordinates (WC) pelos seus limites $x_{\min} \leq x \leq x_{\max}$ e $y_{\min} \leq y \leq y_{\max}$, deverá ser mapeada, simultaneamente, quer no quadrante superior direito, quer no quadrante inferior direito daquele ecrã, maximizando a área disponível, sem deformação e sem recorte do seu conteúdo, devendo os respetivos visores ficar centrados na área disponível.

a) Quais as dimensões dos visores, sabendo que o formato da janela é de 4:3? Não necessita efetuar os cálculos, bastando indicar as respetivas expressões.

b) Indique, através duma composição natural de transformações geométricas 2D, a transformação de enquadramento \mathbf{M} , a usar na forma $\mathbf{p}' = \mathbf{M} \cdot \mathbf{p}$, para o visor no canto superior direito.

$\mathbf{M} =$

c) Qual seria a transformação necessária para implementar uma operação de picking de primitivas gráficas por parte do utilizador no visor referido em b)?

$\mathbf{M}_{\text{pick}} =$

d) Imagine agora que o formato da janela seria 16:10. Quais seriam as novas transformações de enquadramento para os dois visores, por forma a que fossem o mais idênticas possível?

$\mathbf{M}_{\text{topright}} =$

$\mathbf{M}_{\text{bottomright}} =$

4. (3 valores)

Considere as transformações geométricas elementares discutidas nas aulas, usadas na forma $\mathbf{p}' = \mathbf{M} \cdot \mathbf{p}$.

a) para cada uma das seguintes sequências de transformações geométricas 3D, **indique apenas uma composição equivalente**, usando as seguintes regras e prioridades: **1º** uma composição mais simples de transformações geométricas; **2º** uma composição alternativa; **3º** a composição inicial se nenhuma das situações anteriores se aplicar.

1. $S(1,3,4) \cdot S(2,5,1) \cdot T(1,0,0)$ _____
2. $R_x(20^\circ) \cdot R_x(-30^\circ) \cdot T(2,2,2) \cdot R_x(50^\circ)$ _____
3. $S(2,3,3) R_x(20^\circ) R_y(30^\circ) S(1,3,1)$ _____
4. $T(2,5,1) \cdot R_x(270^\circ) \cdot S(1,2,2) \cdot R_x(90^\circ)$ _____

b) Considere uma reta em 2D, definida pela equação $y=mx+b$, sendo m o declive da reta e $(0,b)$ um ponto dessa mesma reta. Pretende-se oferecer, num determinado sistema gráfico, uma operação $\text{mirror}(m, b)$, a qual devolverá a transformação geométrica que efetua uma simetria em relação à reta indicada. Responda, usando uma composição de transformações geométricas 2D elementares (S, R ou T), instanciando os respetivos parâmetros:

$M_{\text{mirror}} =$

5. (4 valores)

a) Considere um ponto P, em 3D, cujas coordenadas homogêneas são $(2, -2, 4, 1)$. Preencha a tabela abaixo com as **coordenadas 3D** da imagem do respetivo ponto, após a projeção indicada:

Alçado Principal	Planta	Alçado Lat. Esquerdo	Alçado Lateral Direito

b) Considere os seguintes pontos adicionais $Q=(2, -2, 0, 1)$, $R=(3, -2, 0, 1)$ e $S=(6, 0, 0, 3)$, para além do ponto P da alínea a). Preencha a tabela abaixo, indicando o comprimento e a orientação (ângulo formado com o eixo horizontal do plano de projeção) para cada um dos segmentos de reta indicados, após uma projeção oblíqua com os seguintes parâmetros: (encurtamento) $l=0.4$, (ângulo de fuga) $\alpha=30^\circ$:

Segmento	Comprimento	Orientação
PQ		
QR		
QS		

RS		Não preencher esta entrada.
----	--	-----------------------------

c) se, na projeção referida em b) o valor do fator de encurtamento fosse unitário, $l=1$, que poderia dizer relativamente ao ângulo que as projetantes fazem com o plano de projeção? Teria aumentado ou diminuído? Justifique!

6. (3 valores)

Considere os métodos estudados nas aulas para a remoção de superfícies ocultas:

a) indique uma vantagem e uma desvantagem da utilização do método conhecido por *back face culling* em relação à utilização do algoritmo de *z-buffer*.

b) Admitindo que a cena a visualizar é composta por vários milhares de pequenos polígonos, haverá alguma ordem para o seu desenho capaz de acelerar a sua visualização?
