

N.º _____ Nome _____

Duração da prova: 45 minutos

Cotação de cada pergunta: assinalada com parêntesis rectos

Perguntas de escolha múltipla: cada resposta incorrecta desconta 1/3 do valor da pergunta

Parte Teórica

10%

- a. **[3.3]** As vulgares impressoras de jacto de tinta constituem exemplos de dispositivos
- ☒ i. Matriciais
 - ii. Vectoriais
 - iii. Tensoriais
 - iv. Nenhuma das anteriores
- b. **[3.3]** Qual das seguintes matrizes representa o ponto 3D com coordenadas (4.0, -2.0, -1.0)?
- i. $[4.0, -2.0, -1.0, 0.0]^T$
 - ii. $[2.0, -1.0, -0.5, 1.0]^T$
 - ☒ iii. $[8.0, -4.0, -2.0, 2.0]^T$
 - iv. Nenhuma das anteriores
- c. **[3.3]** Qual das seguintes transformações usaria para transformar o objecto A da Figura 1 no objecto B?
- i. Rotação
 - ii. Escalamento
 - ☒ iii. *Shearing*
 - iv. Nenhuma das anteriores

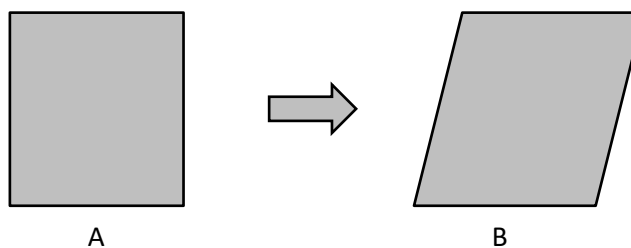


Figura 1

d. [3.3] Que técnica de representação de sólidos está ilustrada na Figura 2?

- i. B-Rep
- ii. CSG
- iii. Octrees
- iv. Nenhuma das anteriores

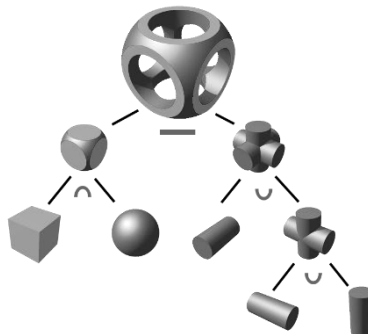


Figura 2

e. [3.3] Nos modelos de iluminação locais, a intensidade da componente de reflexão difusa

- i. Varia consoante a posição da fonte de luz
- ii. Varia consoante a posição do observador
- iii. Pode ser calculada de forma aproximada com o recurso ao vector *halfway*
- iv. Nenhuma das anteriores

f. [3.3] A função de mapeamento de texturas ilustrada na Figura 3 baseia-se numa parametrização

$$x(\varphi, \theta) = \sin \varphi \cos \theta$$

$$y(\varphi, \theta) = \sin \varphi \sin \theta$$

$$z(\varphi, \theta) = \cos \varphi$$

$$\varphi = \pi \cdot t$$

$$\theta = 2\pi \cdot s$$

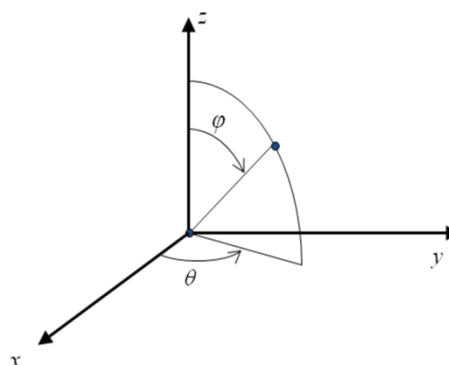


Figura 3

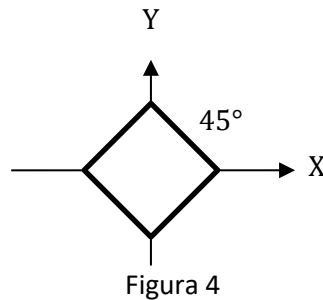
- i. Cúbica
- ii. Cilíndrica
- iii. Esférica
- iv. Nenhuma das anteriores

N.º _____ Nome _____

Parte Teórico-Prática

20%

- a. **[3.0]** Considere o polígono representado na Figura 4. Este está rodado de 45° em torno do eixo Z. Indique as componentes do vector normal unitário.



Normal: (0.0, 0.0, 1.0) ou (0.0, 0.0, -1.0), pois o enunciado não indica para onde está voltado o lado da frente do polígono

- b. **[3.0]** Considere um automóvel, localizado na posição `car.position.x`, `car.position.y`, `car.position.z`, seguindo na direcção `car.rotation.z`. Pretende-se colocar uma câmara a seguir o veículo, colocada na traseira deste a uma distância `DIST` e a uma altura `ALT`. Indique os argumentos dos seguintes métodos usados para configurar a câmara.

```
camera.position.set( car.position.x - DIST * Math.cos(car.rotation.z),
                    car.position.y - DIST * Math.sin(car.rotation.z),
                    car.position.z + ALT);
camera.lookAt(car.position.x, car.position.y, car.position.z);
```

- c. **[1.4]** Um objeto *Sprite* é definido usando

- i. *Geometry* e *Material*
- ii. *Light* e *Scene*
- iii. Apenas *Geometry*
- iv. Apenas *Material*

- d. **[1.4]** Numa *PerspectiveCamera*, o valor do parâmetro *Near*

- i. Pode assumir qualquer valor real
- ii. Tem de ser *Near* < 0.0
- iii. Tem de ser *Near* > 0.0
- iv. Tem de ser *Near* > *Far*

- e. **[1.4]** No Three.js, a orientação da face da frente de um polígono

- i. É definida usando o parâmetro *frontFace*
- ii. Fica sempre orientada para o z positivo
- iii. É definida pela ordem em que se indicam os vértices do polígono
- iv. Não existe, pois os polígonos não têm frente nem verso

N.º _____ Nome _____

- f. **[1.4]** No Three.js, ao usar uma *DirectionalLight*, o cálculo das sombras é efectuado
- ☒ i. Usando uma câmara *Orthographic*
 - ii. Usando uma câmara *Perspective*
 - iii. Usando 6 câmaras *Perspective*
 - iv. Sem o recurso a câmaras auxiliares
- g. **[1.4]** Quando se usa uma *SpotLight*, que propriedades se utilizam para definir a orientação e abertura do foco?
- i. *penumbra* e *decay*
 - ii. *opening* e *orientation*
 - iii. *sweep* e *position*
 - ☒ iv. *angle* e *target*
- h. **[1.4]** Para se realizar o mapeamento de texturas no Three.js, usa-se o atributo da *BufferGeometry*
- i. *texture*
 - ☒ ii. *uv*
 - iii. *st*
 - iv. *texmap*
- i. **[1.4]** No Three.js, se se pretender aplicar numa textura o resultado de uma renderização
- ☒ i. Deve-se usar o *WebGLRenderTarget*
 - ii. Deve-se usar o *RenderTargetImage*
 - iii. Deve-se usar um *Sprite*
 - iv. Não é possível
- j. **[1.4]** Para implementar *picking* no Three.js, é necessário usar
- i. As coordenadas normalizadas do rato (ou local onde se pretende fazer *picking*)
 - ii. A câmara que foi usada na renderização
 - iii. Os objectos aos quais se pretende fazer *picking*
 - ☒ iv. Todas as anteriores
- k. **[1.4]** Quando se activa o *Fog* no Three.js
- i. Todo o *canvas* é afectado
 - ☒ ii. Apenas as zonas do *canvas* com objectos é afectada
 - iii. O *Fog* só afecta os objectos que são filhos do objecto *Fog*
 - iv. Nenhuma das anteriores
- l. **[1.4]** Na técnica de pós-processamento usando *EffectComposer*, a primeira passagem é tipicamente realizada usando
- i. *InitPass*
 - ii. *StartPass*
 - iii. *GlitchPass*
 - ☒ iv. *RenderPass*