



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Computação Gráfica
AP2 - 2º semestre de 2016.

Nome –

Assinatura –

Observações:

- i) Prova sem consulta e sem uso de máquina de calcular.
- ii) Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
- iii) Você pode usar lápis para responder as questões.
- iv) Ao final da prova devolva as folhas de questões e as de respostas.
- v) Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

Na última página encontra-se a folha de respostas. Preencha corretamente e sem rasuras. Todas as questões tem o mesmo peso.

- 1) Seja I a intensidade da luz incidente em um ponto p , l o vetor que indica a direção de incidência da luz, v a posição do observador, n a normal em p e r o raio de luz refletido. NÃO podemos afirmar sobre a equação $(r \cdot v)^t$
 - A Trata-se da componente especular do modelo Phong
 - B Quanto maior x , mais distante o material está de um espelho
 - C não são todos os materiais que possuem esta componente
 - D Esta componente é variante de acordo com a posição do observador
 - E r e v devem estar normalizados para que esta equação seja válida
- 2) As coordenadas de texturas usadas em malhas 3D :
 - A São vértices colocados de forma artificial
 - B Devem ser criadas durante o pipeline gráfico
 - C São coordenadas bidimensionais, atreladas a um vértice
 - D São usadas apenas em texturas procedurais
 - E São geradas depois do estágio de iluminação
- 3) Se não fosse pelo Z-Buffer:
 - A Não poderíamos realizar a interpolação das cores dos vértices.
 - B Faltariam informações para o clipping
 - C Não haveria como estimar quais polígonos estão fora do frustum da camera
 - D Não seria possível pintar os polígonos na ordem de profundidade
 - E Não seria possível aplicar cullings

- 4) Qual destes **elementos não** são **necessários** para calcular o **frustum culling** (por exemplo, a BSP):
- A posição da câmera
 - B Lista de vértices da malha
 - C Lista de **Fontes de Luz**
 - D Direção da camera
 - E Angulo de abertura da camera.
- 5) **Não podemos dizer** que um **vertex shader**:
- A pode ser programado.
 - B Pode gerar novos triângulos
 - C altera os vértices da geometria.
 - D Pode conter um modelo de iluminação
 - E permite manipular coordenadas de texturas
- 6) Um **kernel** é:
- A Uma **função** que é **executada na GPU**
 - B Outro nome dado ao polígono
 - C Um modelo de interpolação
 - D Um shader
 - E Um modelo de iluminação
- 7) Usamos **Quaternions** porque:
- A São ótimos algoritmos de iluminação
 - B Para resolver problemas de profundidade, na etapa de projeção
 - C Para realizar clipping de polígonos
 - D Para construir as matrizes de transformação
 - E No lugar da **matriz de rotação**, para **evitar erros acumulados**
- 8) Podemos dizer que as **texturas procedurais**:
- A São texturas que correspondem a uma imagem bitmap estática
 - B são funções que em geral **não contém periodicidade de padrões**
 - C São texturas sempre usadas para aplicar rugosidade nas superfícies
 - D São usadas para substituir o reflexo no tempo real
 - E São métodos de anti-aliasing para imagens
- 9) **Não podemos dizer** que o **bump-mapping**:
- A cria deformações aparentes na superfície
 - B não deforma a malha
 - C Precisa de um mapeamento de textura
 - D Podem ser usadas em tempo real
 - E **Não permite** o uso do **componente especular**
- 10) Podemos afirmar que a função **glSwapBuffers(...)** **permite**
- A apagar os bits de profundidade do *framebuffer*.
 - B inverter os canais de cores do *framebuffer*.

C **trocar o conteúdo** atual do **frontbuffer** pelo do **backbuffer**, fazendo com o que o desenho feito no **backbuffer** seja exibido na tela.

D o uso de um único buffer (single buffer) no processo de visualização.

E combinar o conteúdo do **frontbuffer** com o **backbuffer**.

11) A **custo computacional** do algoritmo de **Raytracing** **não** depende da:

A quantidade de polígonos na cena.

B número de pixels na imagem final a ser gerada.

C quantidade de níveis de recursão das reflexões e transmissões de raios no algoritmo.

D quantidade de fontes luz na cena.

E do **ângulo** que as **fontes de luz** fazem com a cena.

12) Assinale a **função da OpenGL** que **permite definir os parâmetros** intrínsecos de uma **câmera sintética**

A **glLookAt(...)**.

B **glTranslate(...)**.

C **glRotate(...)**.

D **glBegin(...)**.

E **glFrustum(...)**.

Tabela de respostas. Preencha sem rasuras apenas uma resposta:

Questão	1	2	3	4	5	6	7	8	9	10	11	12
Resposta	B	C	D	C	B	A	E	B	E	C	E	E