



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância  
**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Computação Gráfica**  
**AP2 - 2º semestre de 2017.**

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) sobre o **ZBuffer NÃO** podemos afirmar
  - A Tem a mesma resolução do Frame Buffer
  - B Somente é usado quando não há **Back Buffer** presente
  - C Funciona para resolver problemas de sobreposição de pixels
  - D Deve ser inicializado com valores altos ou baixos, dependendo da convenção
  - E Reside na GPU
- 2) Podemos dizer que utilizamos o **glBegin()** no **OpenGL** para
  - A Inicializar um programa OpenGL
  - B Iniciar o envio de vértices
  - C Inicializar o call-back de desenho
  - D Limpar a tela no término de um frame
  - E Criar um novo material a ser usado
- 3) O **occlusion culling** de polígonos consiste em:
  - A projetar o polígono no frustrum da camera
  - B rasterizar o interior do polígono
  - C recortar parte do polígono que ficou fora da área de projeção
  - D estratégia de eliminar polígonos não visíveis
  - E sinônimo de ZBuffer
- 4) No **OpenGL**, a **normal** de um **polígono** é:
  - A Guardada no vértice

- B Guardada no material do objeto
  - C Guardada na textura do objeto
  - D A mesma para todo o objeto
  - E Calculada de acordo com a posição da camera
- 5) Sobre o **CUDA**, podemos afirmar que:
- A é a linguagem usada para programar shaders
  - B é uma das memórias da GPU
  - C Implementa todo o pipeline gráfico
  - D é uma API gráfica
  - E Permite **acesso a Gpu** para **cálculos não gráficos**
- 6) 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**
- 7) **Não** é uma **estrutura de dados** para **geometria**
- A Quadtree
  - B Octree
  - C BPS
  - D Triangle Fans
  - E **Cohen-Sutherland**
- 8) A **componente de iluminação especular** num ponto  $p$  pode ser descrita no **Phong** como
- A  $Cor(p) = material \cdot (N.L)$ , sendo  $N$  a normal do ponto e  $L$  o vetor de luz para o mesmo ponto
  - B  $Cor(p) = material \cdot (N.O)$ , sendo  $N$  a normal do ponto e  $O$  o vetor de observador para o mesmo ponto
  - C  $Cor(p) = material \cdot L$ , sendo  $L$  o vetor de iluminação
  - D  $Cor(p) = material \cdot (R.O)^n$ , sendo  $R$  o **vetor de reflexo**,  $O$  o **vetor do observador** para o mesmo ponto e  $n$  um **coeficiente do material**
  - E  $Cor(p) = material \cdot I$ , sendo  $I$  a constante da luz da cena
- 9) Quando realizamos **transformações geométricas** no espaço, usando **ângulos de Euler**, podemos ter como **problema** o **Gimbal lock**, que consiste em:
- A falta de ajustes adequados dos parâmetros
  - B **Perda** de **graus de liberdade** em certas configurações
  - C Transformações que anulam outras transformações
  - D Ambiguidade de rotação
  - E Ausência de dados na matriz
- 10) Podemos dizer que o **Environment-mapping**:
- A É uma das etapas do Ray-tracing

- B É uma textura para deformar uma geometria
- C **Simula** o **reflexo** na rasterização
- D É uma técnica de culling de terrenos
- E Faz com que um material pareça estar emitindo luz

11) **Malhas de terrenos** podem ser bastante **extensas** e **consumir** bastante **tempo** de **rendering**.

Para otimizá-los, podemos

- A Usar pixel shaders
- B Iluminar apenas alguns de seus vértices
- C Criar uma amostragem estatística
- D Usar **Level Of Details**
- E Usar Portais

12) Porque o **bump-mapping** precisa de **shaders** para ser implementado em **tempo real**?:

- A precisa ser **aplicado pixel a pixel**
- B É necessário tratar o clipping antes de ser aplicado
- C Precisa de uma textura procedural
- D precisa implementar o Ray-tracing
- E Porque apenas é aplicado no componente especular

**Tabela de respostas. Preencha sem rasuras apenas uma resposta:**

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