



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Computação Gráfica
AP3 - 1º semestre de 2019.

Nome –

Assinatura –

Observações:

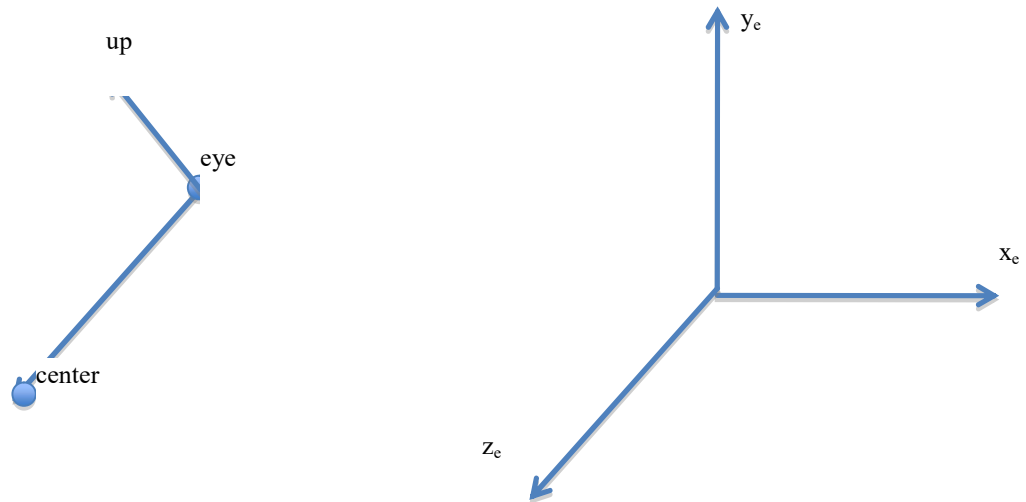
- i) Prova sem consulta e sem uso de máquina de calcular.
 - ii) Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 - iii) Você pode usar lápis para responder as questões.
 - iv) Ao final da prova devolva as folhas de questões e as de respostas.
 - v) Todas 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) Porque o Raytracing não precisa de ZBuffer?:
 - A A ordem de renderização já é feita de trás para frente
 - B O cálculo de profundidade é feito automaticamente quando se realiza o cálculo de interseção com os elementos da cena
 - C A profundidade é garantida pelo pixel shader
 - D Apenas se calcula a iluminação de partes visíveis da cena
 - E Não ocorre sobreposição de objetos
- 2) Um kernel, em CUDA, 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
- 3) Um programador de jogos desenvolveu uma técnica para evitar que árvores não visíveis pela câmera sejam enviadas para a GPU. Pode-se dizer que o desenvolvedor criou uma técnica de:
 - A Clipping
 - B vertex shader
 - C Culling
 - D Física
 - E Transformação geométrica

- 4) A diferença de um pixel shader para um vertex shader é:
- A são processados na Gpu
 - B são gerenciados pela API gráfica
 - C operam em estruturas de dados diferentes
 - D estão no mesmo estágio do pipeline
 - E Requerem uma malha regular
- 5) 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.
- 6) APIs de aceleração do ray-tracing nas GPUs podem se beneficiar do alto paralelismo das mesmas, pois
- A O cálculo de raios lançados para cada região da cena possuem bastante independência e portanto são trivialmente paralelizáveis
 - B A iluminação pode ser calculada de forma mais rápida que na rasterização
 - C O Culling é feito de forma trivial
 - D A recursão pode ser facilmente paralelizada
 - E não é necessário o ZBuffer
- 7) Determine aquela que **não** é uma propriedade de uma curva Bézier Cúbica.
- A A curva se restringe ao fecho convexo uma vez que as funções de base somam 1 (um) para todo valor de u .
 - B Os pontos de controle não exercem controle local pois, mover um ponto de controle move toda a curva (As funções de base são diferentes de 0 em todo o domínio exceto em $u=0$ e $u=1$).
 - C Os vetores tangentes à curva nos pontos extremos coincidem com a primeira e última aresta do polígono de controle.
 - D A curva não oscila sobre nenhuma reta mais do que oscila o polígono de controle (*propriedade de minimização de variação*).
 - E A curva não pode ser transformada por *transformações afins* (translações e rotações) definidas sobre os pontos de controle. É preciso aplicar a transformação em todos os pontos da curva.

- 8) Dada uma posição de uma câmera $eye = (eye_x, eye_y, eye_z)$, um centro de visada $center = (center_x, center_y, center_z)$ e um vetor que indique a direção para cima da câmera $up = (up_x, up_y, up_z)$, um sistema de coordenadas de câmera dado pelos vetores x_e , y_e e z_e pode ser definido através das seguintes equações



- A $x_e = \frac{up \times z_e}{\|up \times z_e\|}, y_e = \frac{z_e \times x_e}{\|z_e \times x_e\|}, z_e = \frac{(center - eye)}{\|center - eye\|}$
- B $x_e = \langle up, z_e \rangle, y_e = \langle x_e, y_e \rangle, z_e = \frac{(center - eye)}{\|center - eye\|}$
- C $x_e = \frac{z_e \times up}{\|z_e \times up\|}, y_e = \frac{x_e \times y_e}{\|x_e \times z_e\|}, z_e = \frac{(center - eye)}{\|center - eye\|}$
- D $x_e = \frac{z_e \times up}{\|z_e \times up\|}, y_e = \frac{x_e \times y_e}{\|x_e \times z_e\|}, z_e = \frac{(eye - center)}{\|eye - center\|}$
- E $x_e = \frac{up \times z_e}{\|up \times z_e\|}, y_e = \frac{y_e \times x_e}{\|y_e \times x_e\|}, z_e = \frac{(center - eye)}{\|center - eye\|}$

- 9) **Não** é uma propriedade de uma interpolação bilinear:

- A É usada para interpolar funções de duas variáveis.
- B Para uma dada amostra, a função de interpolação resultante é uma função quadrática.
- C Utiliza os valores das quatro amostras vizinhas mais próximas para definir a interpolação em um ponto do plano.
- D Ao ser generalizada para 3d recebe o nome de interpolação trilinear.

E Quando usada para ampliar (*zoom in*) uma imagem, causa distorções maiores que o algoritmo de interpolação pelo vizinho mais próximo (*nearest-neighbor-interpolation*).

10) Um sólido **não** pode ser representado por:

- A Um conjunto de voxels.
- B Uma função implícita dada por campo $F(x,y,z)$ de valores densidade.
- C Um modelo B-rep.
- D Uma coleção de tetraedros formando uma triangulação tridimensional.
- E Uma malha de triângulos 2d.

11) Uma transformação afim preserva:

- A Distâncias.
- B Curvas.
- C Ângulos.
- D Retas.
- E Áreas.

12) Um dado de medicina obtido por tomografia computadorizada é um exemplo de objeto gráfico:

- A espacial e bidimensional.
- B planar e bidimensional.
- C espacial e unidimensional.
- D espacial e tridimensional.
- E planar e tridimensional.

Tabela de respostas. Preencha sem rasuras apenas uma resposta:

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