

## Exercícios

### Computação Gráfica Avançada

1. Explique o que é e duas vantagens de utilização da técnica de Vertex Array.
2. Explique o que é e duas vantagens de utilização da técnica de Display Lists.
3. Explique os passos necessários para exibição de informação textual (por exemplo o FPS) sobre uma cena 3D.
4. Explique o que é a técnica de mipmap.
5. Quais são os 3 estágios do pipeline de renderização e operações que cada um realiza?
6. Quais buffers estão dentro do framebuffer? Comente o uso de cada um deles.
  
7. Explique como pode ser verificado que o estágio da aplicação é o gargalo?
8. Explique como pode ser verificado que o estágio de geometria é o gargalo?
9. Explique como pode ser verificado que o estágio de rasterização é o gargalo?
10. Explique como pode ser otimizado o estágio da aplicação?
11. Explique como pode ser otimizado o estágio de geometria?
12. Explique como pode ser otimizado o estágio de rasterização?
  
13. Em relação a um grafo de cena, porque somente as classes Scene e Transform derivam de Group? O que vem a ser a classe group?
14. Qual a vantagem da utilização de um grafo de cena?
  
15. Comente sobre os dois tipos de LOD.
16. Explique o que é o T-vertice, que efeito pode causar na malha e o que deve ser feito para evitá-lo.
17. Qual a vantagem de se utilizar LOD para geração de cenários e personagens?
18. Qual a influência do uso de LOD no uso de CPU. E GPU? Comente.
  
19. Explique como a interpolação linear é utilizada para fazer animação de personagens em formato MD2.
  
20. Explique uma forma de calcular a interseção entre linhas (segmento de reta) em 2D por meio de equações paramétricas.
21. Explique uma forma de calcular a interseção entre uma linha e um círculo em 2D por meio de equações paramétricas.
  
22. Pesquise estratégias para simular em um jogo:
  - Espelho;
  - Sombra;
  - Portais.
- 23.

## 2. Inteligência Artificial para Jogos

1. Comente as principais características das FSM, suas deficiências e como as FSM baseadas em pilha podem resolver as limitações das FSM. Dê um exemplo que mostre vantagem do uso de FSM com pilha.
2. O que é uma FSM? O que é uma FSM com pilha. O que é uma FSM hierárquica. Compare o poder de processamento dos 3 tipos de pilhas.
3. Explique como pode ser implementado um esquema de troca de mensagens de um jogo multiplayer em rede. Para esse mesmo jogo, como pode ser implementado um esquema de troca de mensagens local para garantir que as mensagens enviadas por ações simultâneas entre dois personagens sejam processadas com a mesma prioridade (problema discutido em aula).
4. Em um jogo multiplayer, explique em detalhes (algoritmo abstrato) como pode ser implementado um esquema de troca de mensagens para contabilizar o número de mortes que cada NPC realizou. Considere o problema tratado em aula do envio de mensagens simultâneas entre dois personagens.
5. Comente sobre 3 tipos de grafos de navegação, com aplicações, vantagens e desvantagens.
6. No jogo Duke Nuken (FPS), o tiro da RPG (Bazuca) tem um comportamento de “tentar” seguir o alvo (*steering behavior*). Por meio de figuras e algoritmos, comente como essa arma pode ser implementada em um jogo FPS, em relação à:
  - behaviors utilizados;
  - cálculos de vetores (deslocamento), geometria utilizada e colisão;
  - modelagem do tiro;
  - FSM e mensagens;
  - tempo de vida do tiro;
  - path-finding.
7. Em um jogo do tipo FPS, explique o que é e defina uma FSM de pelo menos 4 estados. Defina atributos relevantes para os NPCs. Explique em detalhes (algoritmo abstrato) como pode ser implementado um esquema de troca de mensagens para contabilizar o número de mortes que cada NPC realizou. Os NPCs disparam somente tiros de bazuca que se movem em linha reta. Considere o problema tratado em aula do envio de mensagens simultâneas entre dois personagens.
8. Considerando-se um grafo de navegação do tipo Pontos de Visibilidade, comente sobre: a) o que é este grafo, b) como é gerado, c) como pode ser aplicado sobre ele o algoritmo A\*, e uma vez encontrado a rota, d) como pode-se utilizar um esquema de suavização de caminhos e) juntamente com algum algoritmo de *steering behavior* para movimentar o personagem de forma suave até o destino. De exemplos gráficos.
9. Um NPC de um jogo do tipo FPS (Ex: Quake) é um agente, um programa ou um simples objeto de software? Justifique e faça comentários em relação às discussões e exemplos apresentados em aula sobre Programas X Agentes X Objetos.
10. Explique o que é FSM, tipos e aplicações. Como pode ser implementado um NPC em relação a sua FSM e Pathfinding em um grafo do tipo POV em um jogo FPS para tratar com a abertura de portas em um cenário tipo labirinto. Considere o fato de colocar IA também no cenário (portas inteligentes).
11. Em relação aos algoritmos de Dijkstra e A\*, comente sobre a fórmula  $f' = g + h'$ . Em qual situação os dois algoritmos são iguais. Comente sobre a influência dos valores de  $g$  e  $h'$  no caminho encontrado. Dê duas situações que não se pode utilizar o A\*, justificando.

12. No jogo Duke Nuken (FPS), o tiro da RPG (Bazuca) tem um comportamento de “tentar” seguir o alvo (*steering behavior*). Faça uma modelagem vetorial do deslocamento do missel (vetor posição e cálculo do vetor direção, ângulos, forças), considerando-se o loop de renderização do jogo. Comente sobre a implementação desta entidade, mensagens, tempo de vida do tiro, danos e teste de colisão.
13. Explique o processo de suavização de caminhos. Quando deve ser aplicado?
14. Explique o processo de geração de caminhos pré-calculados.
- 15.

### 3. Programação em GPU

1. Implemente shaders para gerar um *height map* em GPU a partir de uma textura e uma grade regular de vértices.
2. Implemente um fragment shader para fazer a detecção de bordas em imagens. Compare o desempenho com o mesmo algoritmo executando somente em CPU. Faça os mesmos testes para outros efeitos sobre imagens, como por exemplo mudança de saturação, brilho, transformação de RGB em tons de cinza.
3. Implemente iluminação spot em GPU, com atenuação gradativa em relação a abertura.
4. Implemente a técnica de bump-mapping em GPU sobre uma superfície plana. Utilize uma textura como bump-function.
5. Implemente algoritmos para gerar texturas procedurais em GPUs. Ex: padrões quadriculados (tabuleiro de xadrez), padrões segundo funções trigonométricas, etc.