



Reposição 3

Computação Gráfica - Prof. Bernardo Lima

Turma: CP28CP

Aluno: Guilherme Rodrigues dos Santos

Data: 04/09/2024

1. Renderize um modelo 3d de um cubo da sua cor favorita. Sugiro utilizar three.js para fazer essa atividade.

Em anexo.

2. Implemente a triangulação de delaunay. Seu algoritmo deve receber como entrada um conjunto de pontos e fornecer como saída um conjunto de triângulos. Seu programa também deve renderizar esse conjunto de triângulos usando SVG.

Em anexo.

3. Apresente o pseudocódigo para a abordagem frustum culling.

Loop de Desenho da aplicação

```
{  
    Carregar identidade para a matriz projeção  
    Carregar identidade para a matriz transformação  
  
    Aplica transformações na matriz projeção  
    Aplica transformações na matriz transformação  
  
    Calcula Frustum  
  
    Loop de desenho dos objetos  
    {  
        Se (objeto no Frustum)  
            Desenhando objeto  
    }  
}
```

4. Assista a estes dois vídeos e descreva a abordagem para a criação do toon shader. Inclua o pseudocódigo para o(s) shader(s) que você usaria para chegar a esse efeito.

O toon shader é composto por um vertex shader e um fragment shader. O vertex shader é responsável por transformar as coordenadas dos vértices e as normais, enviando essas informações para o fragment shader. Por sua vez, o fragment shader aplica o efeito de cel

shading, quantizando a iluminação em níveis distintos para criar um efeito de cor sólida característico dos desenhos animados. A iluminação é calculada com base na direção da luz e na normal dos fragmentos, ajustando a cor para representar diferentes intensidades luminosas. Para acentuar o estilo toon, é possível adicionar um efeito de contorno. Isso geralmente envolve desenhar o objeto uma segunda vez com uma leve expansão, criando uma borda ao seu redor. Um shader de contorno aplica uma cor de linha ao redor do modelo, utilizando técnicas de dilatação para realçar as silhuetas e proporcionar um visual mais estilizado e definido.

Toon Shader.

```
{
    // Vertex Shader
    Function VertexShader(VertexInput)
    {
        // Transformar a posição do vértice e a normal
        TransformedPosition = TransformVertex(VertexInput.Position)
        TransformedNormal = TransformNormal(VertexInput.Normal)
        Output = { TransformedPosition, TransformedNormal }
        Return Output
    }

    // Fragment Shader
    Function FragmentShader(FragmentInput)
    {
        // Definir a direção da luz
        LightDirection = SetLightDirection()

        // Calcular o sombreado baseado na normal e na direção da luz
        DotProduct = CalculateDotProduct(FragmentInput.Normal, LightDirection)
        ToonShade = ApplyQuantization(DotProduct)

        // Aplicar o Toon Shader à cor base
        FinalColor = ApplyToonShade(BaseColor, ToonShade)
        Return FinalColor
    }
}
```

5. Cite três aplicações do conceito de frame buffer.

Edição de Vídeo e Pós-processamento:

Em softwares de edição de vídeo e efeitos visuais, o frame buffer é utilizado para armazenar e manipular imagens durante o processo de edição. Ele possibilita a aplicação de efeitos, ajustes de cor e filtros nas imagens de uma sequência de vídeo antes da exportação, como o vídeo final. Isso é fundamental para realizar manipulações precisas e de alta qualidade.

Software de Streaming e Ferramentas de Captura de Tela:

Nos softwares de captura de vídeo e streaming ao vivo, o frame buffer armazena temporariamente as imagens que estão sendo capturadas da tela ou de uma fonte de vídeo. Esse armazenamento temporário é vital para assegurar que o vídeo capturado possa ser processado e transmitido em tempo real com mínima latência e sem perda de qualidade.

Ambientes de Desenvolvimento de Software e Aplicações Desktop:

Nos ambientes de desenvolvimento de software e aplicações desktop, o frame buffer desempenha um papel crucial na renderização e atualização dos elementos da interface gráfica do usuário, como janelas, botões e menus. Isso assegura que as interfaces sejam corretamente exibidas e que as modificações na interface sejam efetivamente refletidas para o usuário.