UNIVERSIDADE FEDERAL DA PARAÍBA

Discente: Carlos Fabrício da Silva Pontes

Discente: Davi José Lucena Luiz

Docente: Christian Azambuja Pagot

Disciplina: Introdução à Computação Gráfica

ATIVIDADE PRÁTICA 5

João Pessoa

2021

1. Desenvolvimento da atividade:

No desenvolvimento do exercício 1 da atividade 5, utilizamos como base o modelo de iluminação de Phong para realizar a extensão do modelo de iluminação do *ray tracer* para que fosse possível suportar o modelo de iluminação de Phong, de forma que fosse incluído o termo especular para a realização desta etapa. Primeiramente, realizamos o implemento do termo especular para que fosse possível gerar a iluminação sobre a esfera que seria renderizada, para isso foi necessário calcular algumas variáveis, R e V, para então calcular o termo especular e incluí-lo no modelo. Para o cálculo da variável r, foi necessário fazer a reflexão do vetor L em um vetor normal, no caso o vetor normal ao ponto de intercessão, e depois normalizar. Já para a variável V foi necessário encontrar o vetor que apontasse para câmera, para isso, pegamos a origem do raio, já que este sai da câmera, e subtraímos do ponto de intersecção. Após alguns cálculos e ajustes de variáveis conseguimos renderizar a esfera como fora solicitado.

Após concluído o primeiro exercício, iniciamos o desenvolvimento do segundo exercício onde é solicitado a implementação de um suporte que seja capaz de renderizar triângulos. Iniciamos a implementação da primitiva realizando a criação da classe triângulo a qual possui três variáveis (a, b, c) que são equivalentes a cada vértice da primitiva. Em seguida, implementamos o Algoritmo de Intersecção Möller-Trumbore, que possui como objetivo calcular a intersecção de um raio e um triângulo que possui 3 dimensões sem a necessidade de uma pré-comparação com a equação do plano contendo a primitiva.

Dentro do método interseccionar, calculamos a normal do plano para aresta AB e para a aresta AC, após isso foi calculado o produto vetorial entre a aresta AC e a direção do raio, calculamos também o produto interno entre a aresta AB e o resultado anterior que foi obtido.

Todo esse passo a passo foi necessário para verificar se o raio é paralelo ao triângulo, e caso essa condição fosse satisfeita o retorno seria “*false*” e o bloco de execução que possui o algoritmo seria interrompido, pois dessa forma não seria possível do raio intersectar o triângulo.

Com o passo a passo anterior concluído continuamos a implementar o Algoritmo de Intersecção Möller-Trumbore, mas dessa vez especificamente para encontrar o ponto de intersecção fazendo uso de uma variável que permite a continuidade da execução do bloco para realizar os cálculos caso sua condição seja satisfeita, caso contrário o seu retorno é “*false*”.

Com o ponto de intersecção encontrado é possível prosseguir com os cálculos, localizando primeiramente a posição do ponto para seja possível o raio iluminá-lo de forma precisa, e em seguida normalizar cada aresta do triângulo afim de obter a sua iluminação.

Após concluída toda esta etapa, criamos uma variável dentro do bloco de função Render, para guardar um objeto Triângulo que já recebe como parâmetro os seus vértices e assim renderizá-lo na tela com a iluminação correta.

2. Estratégias adotadas para a resolução da atividade:

Para o desenvolvimento de cada exercício da atividade 5, utilizamos como base algumas propriedades básicas da álgebra linear como produto vetorial, produto interno e a norma de um vetor. Adotamos o uso de algumas propriedades da iluminação de Phong para a resolução da primeira questão, fizemos uso principalmente do termo especular, o qual foi preciso implementar no desenvolvimento da questão 1 para fins de uso para a iluminação e a coloração do pixel. Já para a segunda questão foi preciso uma abordagem mais específica pois era preciso implementar um algoritmo que fosse capaz de realizar o cálculo da intersecção de um raio com um triângulo de 3 dimensões. Para a resolução desta segunda questão implementamos o Algoritmo de Intersecção Möller-Trumbore, o qual foi sugerido como uma abordagem de resolução da questão. Tal algoritmo é capaz de realizar o cálculo que é solicitado. Ao longo da implementação do algoritmo fizemos uso de algumas propriedades da álgebra linear e utilizamos como base também o artigo que fora sugerido no PDF da atividade 5.

3. Dificuldades:

A principal dificuldade que tivemos ao longo do desenvolvimento desta atividade 5 foi na iluminação de cada primitiva. No início do implemento de cada exercício encontramos dificuldades no cálculo do raio com o ponto de intersecção de cada primitiva, sempre que modificávamos uma parte da equação a iluminação não acontecia como o planejado (Figura 1).

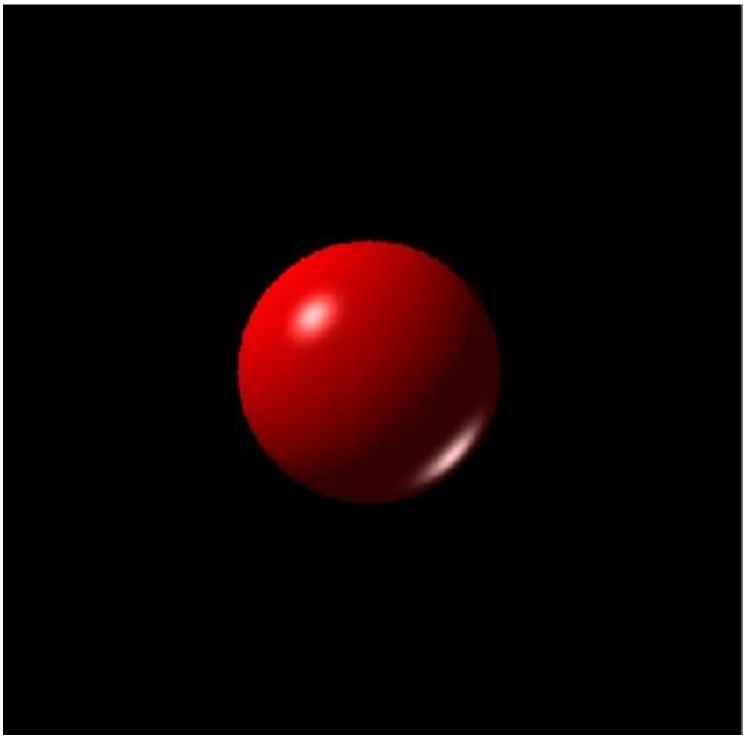

Figura 1 – Esfera iluminada de forma incorreta

Na imagem acima é possível perceber uma pequena iluminação na parte inferior direita da esfera, esse foi um dos erros que tivemos no desenvolvimento da primeira questão.

Além disso, tivemos o mesmo erro no desenvolvimento da questão 2, a qual se trata da primitiva triângulo, passamos por dificuldade no cálculo final do raio com o ponto de intersecção e também no desenvolvimento do algoritmo que foi implementado. Neste quesito o erro principal foi o de iluminação, o triângulo não estava sendo iluminado corretamente e em algumas tentativas estava sendo iluminado no seu ponto central (Figura 2) ou em alguns casos não estava sendo iluminado de forma alguma (Figura 3).

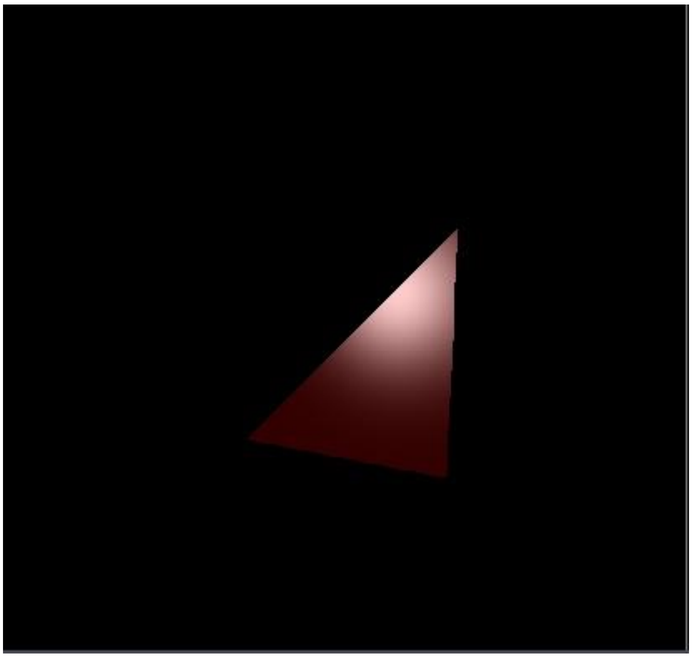

Figura 2 – Triângulo iluminado de forma incorreta e sem coloração

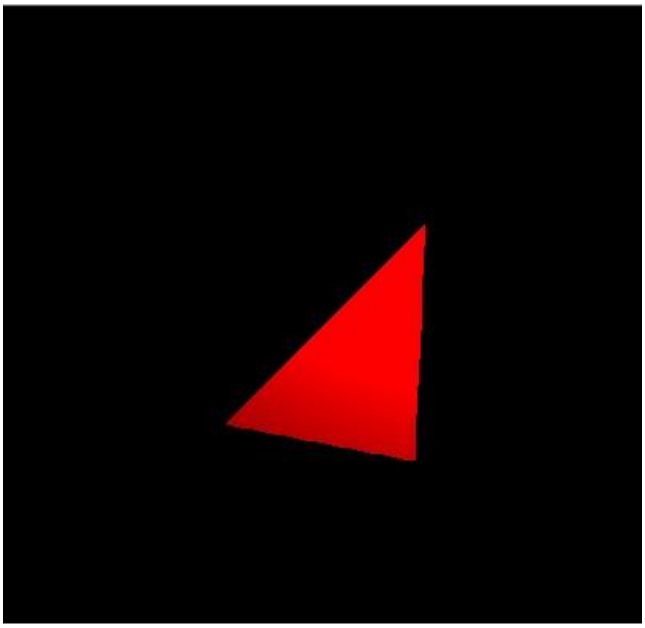

Figura 3 – Triângulo vermelho com pouca iluminação

Em alguns casos tivemos dificuldades também no cálculo da posição do triângulo, encontramos um pequeno erro que localizado no terceiro vértice (Figura 4). Passamos os vértices que foram informados no PDF da atividade, no entanto o triângulo estava em uma posição incorreta e com uma aparência mais fina, em seguida fizemos a mudança da primeira coordenada do terceiro vértice de -0.75 para 0.75 e assim conseguimos renderizar o triângulo com a sua posição correta.

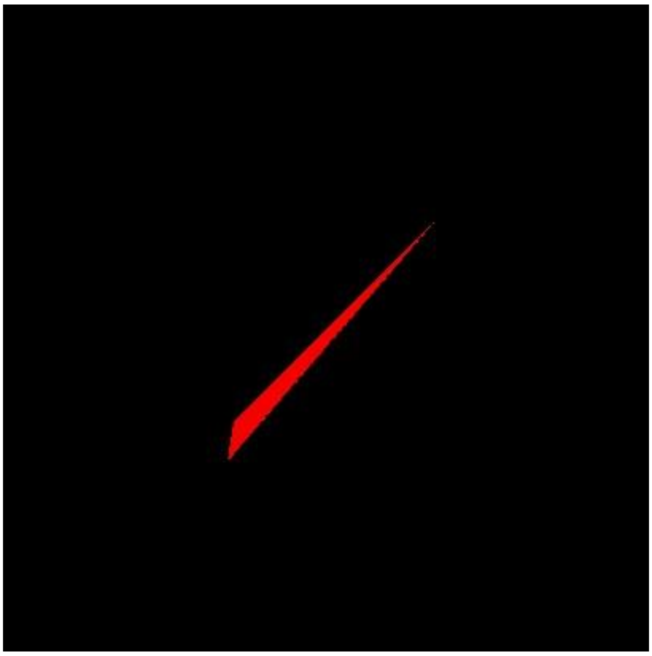

Figura 4 – Triângulo vermelho posicionado de forma incorreta e com uma aparência fina

4. Resultados gerados e soluções:

Após ter passado por alguns erros no cálculo de iluminação das primitivas e também no cálculo das posições, conseguimos desenvolver as resoluções para as duas questões como foram solicitadas, através de propriedades da álgebra linear e da implementação do Algoritmo de Intersecção Möller-Trumbore.

Abaixo segue uma imagem da esfera iluminada, resultado obtido após todo o desenvolvimento da questão 1 (Figura 5).

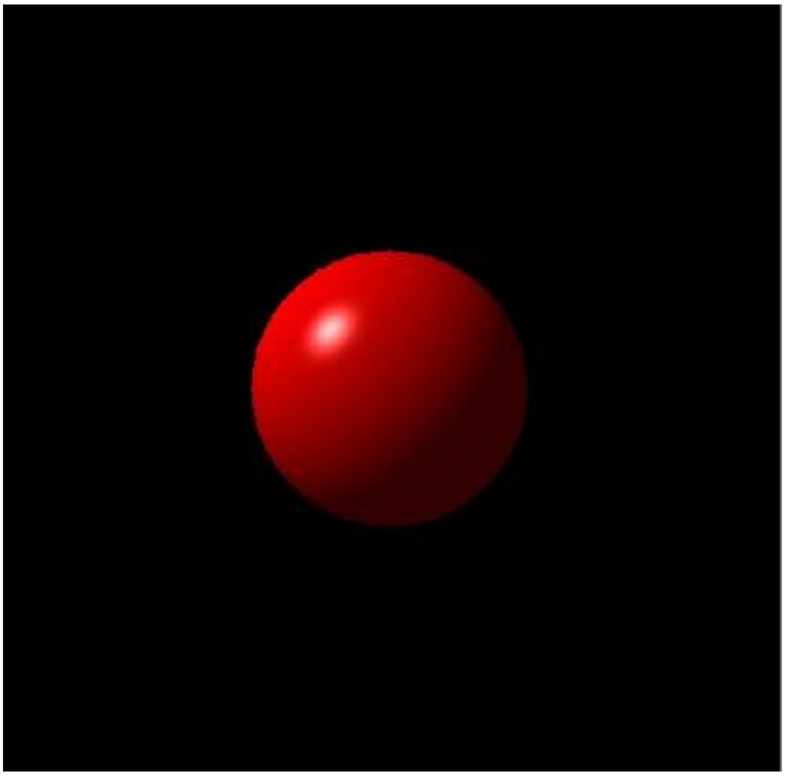

Figura 5 – Esfera vermelha iluminada com o modelo de iluminação de Phong

Abaixo segue uma imagem do triângulo iluminado como foi solicitado, resultado que foi obtido após a implementação do algoritmo e de todo o desenvolvimento que foi necessário para a resolução da questão (Figura 6).

5. Possíveis melhoras:

Dado que a atividade foi realizada utilizando o método mais rápido e o mais indicado, não identificamos nenhuma possível melhora.

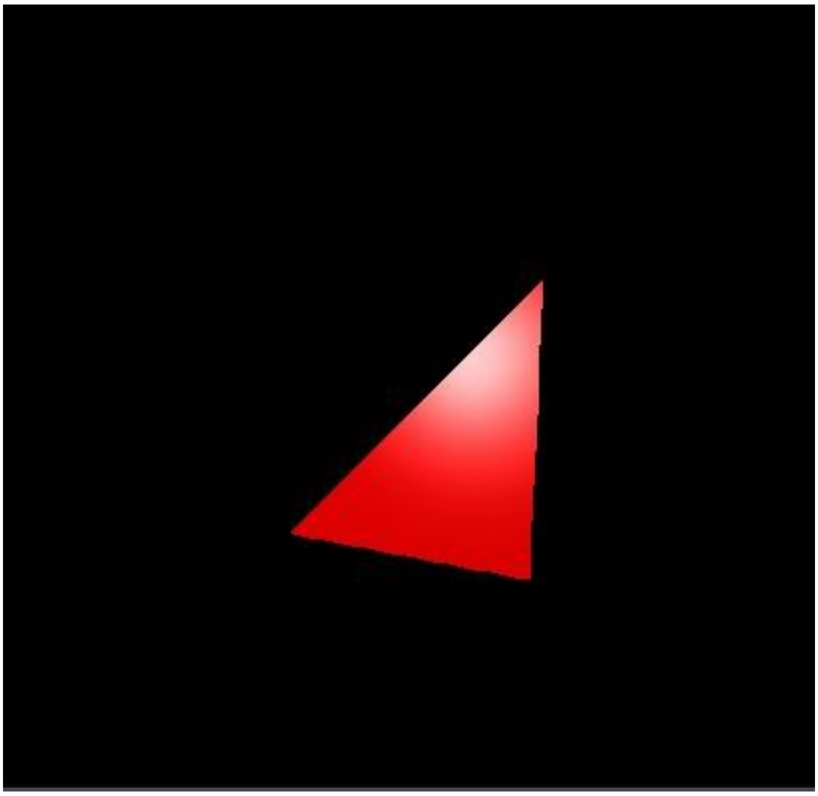

Figura 6 – Triângulo vermelho iluminado com o modelo de iluminação de Phong

6. Referências Bibliográficas:

https://en.wikipedia.org/wiki/M%C3%B6ller%E2%80%93Trumbore_intersection_algorithm

https://cadxfem.org/inf/Fast%20MinimumStorage%20RayTriangle%20Intersection.pdf

https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-rendering-a-triangle/why-are-triangles-useful

7. Link para repositório online codepen da atividade 5:

https://codepen.io/davi-luiz/pen/wvdMpjX?editors=0010

Obs: Para verificar a renderização do triângulo basta apenas descomentar a linha de código que possui a seguinte sintaxe: “let s1 = new Triangulo (new THREE.Vector3(-1,-1,-3.5), new THREE.Vector3(1.0,1.0,-3.0), new THREE.Vector3(0.75, -1, -2.5));”.

E em seguida comentar a linha de código da esfera que se encontra acima da linha de código do triângulo, ou vice-versa.