



**CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA**

Atividade 2

Ana Maria Pinto da Silva Neta - 20160143190

João Pessoa 2020

Introdução

Este trabalho consiste em implementar as técnicas básicas da Rasterização apresentadas na disciplina de Introdução à Computação Gráfica. Serão realizados os seguintes aspectos: a rasterização de um ponto, de uma linha, de um triângulo e cores.

O trabalho foi desenvolvido utilizando uma framework, disponibilizada pelo docente responsável pela disciplina, Christian Azambuja Pagot. A linguagem adotada foi C++.

Desenvolvimento

A rasterização de uma figura é basicamente acender um conjunto de pixels na tela com uma determinada cor, que neste caso, o que seja preferível para o discente. Este conjunto de pixels podem ser combinados cada um com uma respectiva onde dará forma, e cor para a figura que será escolhida.

Para o desenvolvimento desta atividade alguns requisitos deverão ser atingidos, são esses:

- Implementar a interpolação de cores: a partir de dois pontos, a operação de interpolação deverá interpolar a cor da linha com base nas dimensões e cores pré-estabelecidas pelo discente.
- Implementar a função PutPixel : que deverá receber como entrada as coordenadas de um ponto e uma cor em RGBA, para que possamos fazer a rasterização de um determinado ponto na memória de vídeo.
- Implementar a função DrawLine : responsável por rasterizar uma linha, com dois pontos sendo atribuídos como entrada para um respectivo vértice. A reta resultante será dada a partir da interpolação e sua coloração se tornará um degradê.
- Implementar a função Draw Triangle: responsável em receber como parâmetros três vértices e suas respectivas cores, para que este irá chamar a função DrawLine e um triângulo possa ser plotado.

PutPixel

A tela de um computador pode ser representada matematicamente como um plano de duas dimensões, com coordenadas x e y , onde um ponto nesta tela poderá ser representado pelo ponto $P = (X_p, Y_p)$, sendo assim o Pixel seria este ponto.

Então, se o nosso objetivo é desenhar um ponto em nossa tela, devemos informar as coordenadas que desejamos e a cor que queremos que ela seja.

Foi proposto para essa atividade que através do método PutPixel(), será informado os parâmetros de coordenadas e a cor, esta cor precisa está no formato RGBA(Red, Green, Blue, Alpha). Com isso podemos obter uma maneira de encontrar o endereço da memória relacionada a cor de cada pixel:

$$end = p.x * 4 + p.y * 4 * IMAGE_WIDTH$$

onde *IMAGE_WIDTH* corresponde à largura da tela.

Para implementá-lo, precisamos criar um struct Pontos para que possamos definir um ponto, sendo as coordenadas x e y, e também foi criado um struct Cor, que terá os valores, correspondentes às bandas R,G,B e A.

DrawLine

Para que este método seja implementado foi utilizar o algoritmo do ponto médio, o docente o recomendou e demonstrou nas aulas sua forma simplificada. Alguns passos foram realizados:

- Primeiro, temos variáveis auxiliares para ajustar os valores quando as coordenadas do pixel final forem menos do que a do pixel inicial;
- Utilizou-se um uma instrução condicional para a verificação se o coeficiente angular é menor que 1, caso seja maior apenas invertemos as coordenadas;
- Por fim, a implementação da interpolação linear.

DrawTriangle

Por fim, o método restante a ser implementado, que a partir de que temos a capacidade de rasterizar uma linha, sem o preenchimento de cores, este método basicamente chama o método acima para rasterizar linhas com o objetivo de formar a figura de um triângulo.

Resultados

Estes foram os resultados obtidos a partir da implementação.

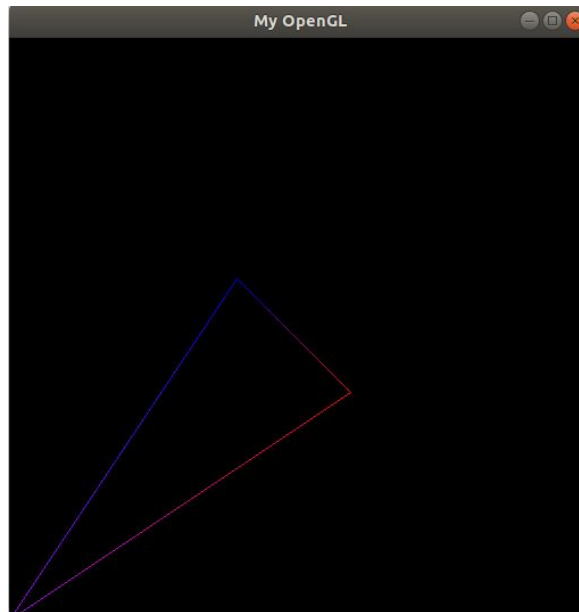


Figura 1 - Teste da função Drawtriangle.

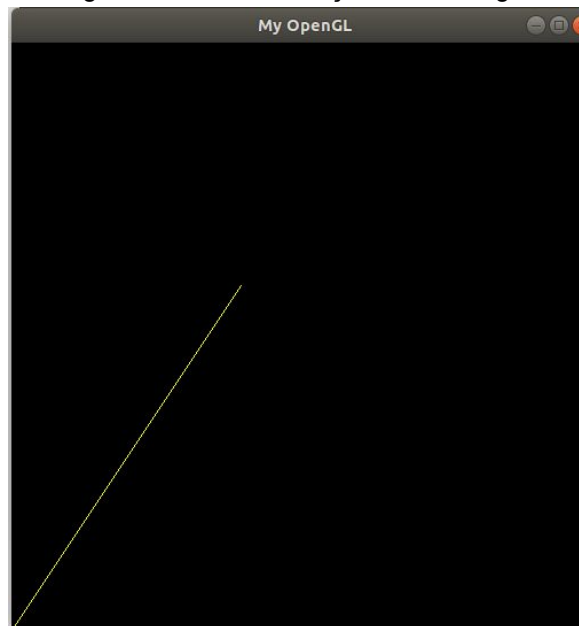


Figura 2 - Teste da função DrawLine..

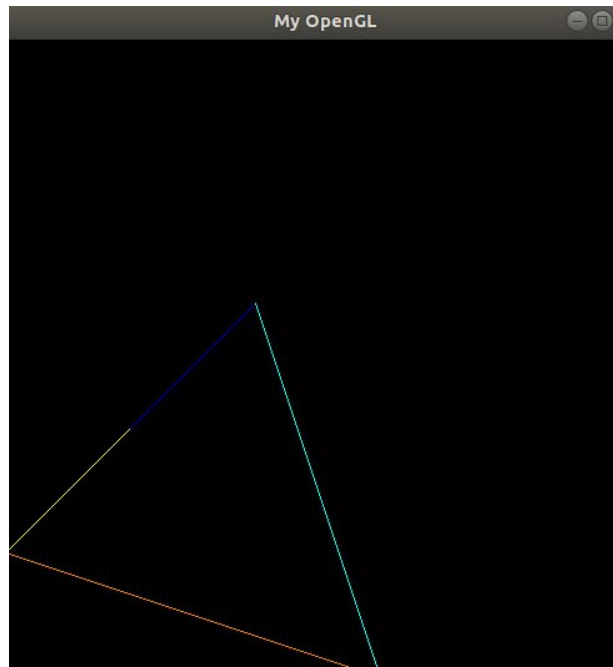


Figura 3 - Teste da função Drawtriangle

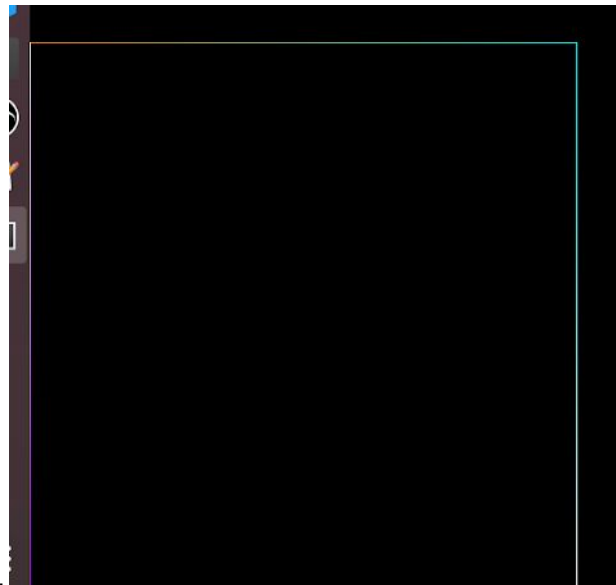


Figura 4 - Teste da função Drawline.

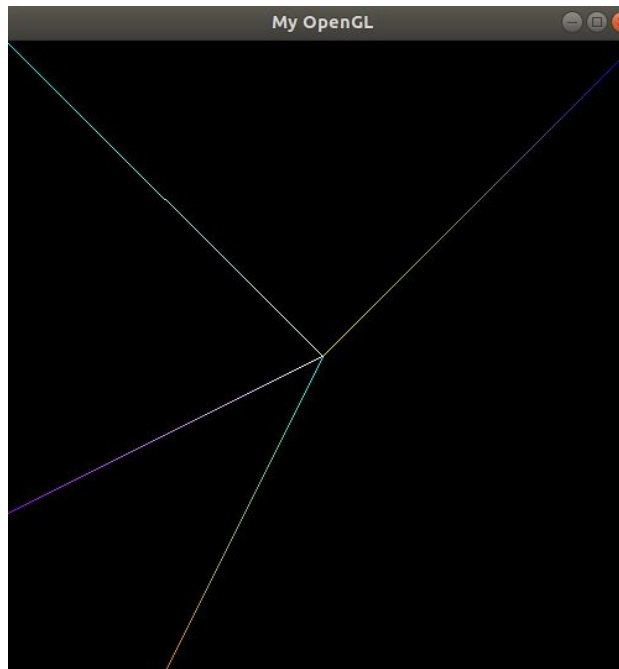


Figura 5 - Teste da função Drawline.

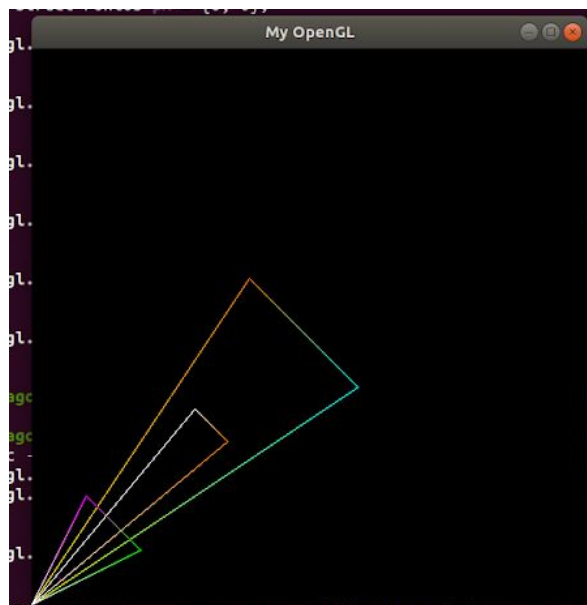


Figura 6 - Teste da função Drawline.

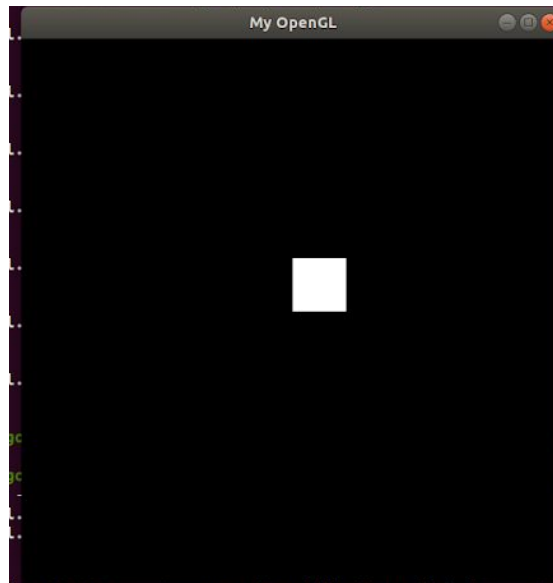


Figura 6 - Teste da função PutPixel..

Referências

- Aulas no youtube do docente Christian A. Pagot
- <http://www.sunshine2k.de/coding/java/TriangleRasterization/TriangleRasterization>
- http://www-users.mat.uni.torun.pl/~wrona/3d_tutor/tri_fillers.