

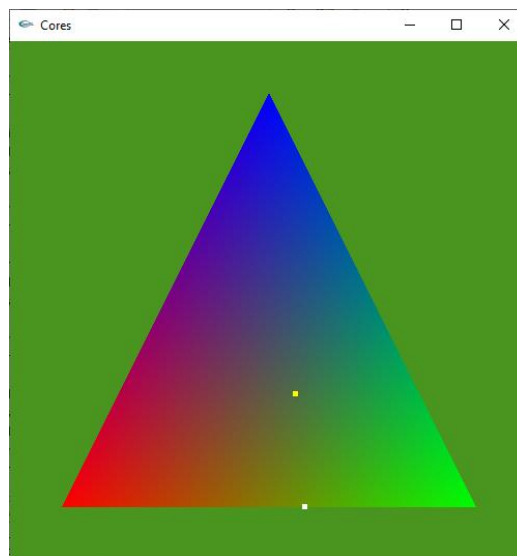
Exercício – Interpolação de Cores

Objetivo: Trabalhar os conceitos básicos interpolação de cores.

Material: Sistema operacional Linux (recomendação Ubuntu 20.04) com g++, OpenGL e GLUT instalados (geralmente só são necessários dois comandos para a instalação, <https://gist.github.com/AbdullahKady/f2782157991df652c2baee0bba05b788>).

1. Tarefa:

Criar uma aplicação “Cores!” de computação gráfica. A aplicação cria uma janela com um triângulo colorido na tela, como na figura abaixo. Cada vértice do triângulo possui uma das três cores primárias, vermelho, verde e azul. A cor de fundo é o resultado da escolha do usuário ao clicar dentro do triângulo. Para isso, deve-se interpolar as contribuições das três cores primárias apropriadamente. O funcionamento básico esperado para a aplicação pode ser visto no vídeo anexo. O usuário pode clicar na tela e escolher a cor utilizando o botão esquerdo do mouse e pode alterar o triângulo arrastando seus vértices com o botão direito do mouse. Exemplo do funcionamento esperado pode ser visto no vídeo “App - cores.mp4”.



1.1. Passo: aplicação base

- Compilar o arquivo “cores.cpp” fornecido com este roteiro com o comando “g++ -o test cores.cpp -lGL -lGLU -lglut”. As bibliotecas OpenGL, GLU e GLUT são referenciadas respectivamente com os comandos -lGL -lGLU -lglut.
- Executar o arquivo compilado com o comando “./test”
- Verificar se uma janela foi criada como descrito para esta tarefa
- Verificar se um ponto é criado ao clicar com o botão esquerdo na janela
 - O ponto deve acompanhar o mouse enquanto o botão estiver clicado
- Verificar se é possível modificar o triângulo arrastando seus vértices com o botão direito do mouse
- Nos passos abaixo, modificar a função *motion*
 - Parte indicada com “COLOQUE SEU CODIGO AQUI”

1.2. Passo: calcular ponto projetado

- Calcular a posição do ponto clicado após projetá-lo na reta formada pelos pontos vermelho e verde
 - A projeção deve considerar a interseção da reta formada pelo ponto azul e o ponto clicado e, a reta formada pelos pontos vermelho e verde
 - Não precisa tratar as extrapolações dos limites do triângulo, ou seja, pode extrapolar
- Atualizar as variáveis pProjX e pProjY

1.3. Passo: calcular a cor de fundo interpolada

- Calcular a cor de fundo interpolada considerando a posição clicada pelo usuário
 - A cor final deve considerar as contribuições das três primárias
 - Quanto mais perto de uma primária, mais ela contribui
 - Utilizar o ponto projetado para ajudar a interpolar a cor final
 - Dica: ponderar entre azul e o resto e depois ponderar entre vermelho e verde considerando o que já foi para o azul
 - Não precisa tratar o comportamento de cliques fora do triângulo, ou seja, deixar interpolar o valor que der
- Atualizar as variáveis que representam a cor de fundo gR, gG e gB (vermelho, verde e azul respectivamente)