

Prática 2 – Computação Gráfica – OpenGL

Teoria

As coordenadas da tela são normalizadas, isso significa que os valores das posições x e y variam entre -1 e 1, sendo (0,0) a posição central. Quando recebemos as coordenadas do mouse (x,y) primeiro é necessário normalizá-las e depois atualizar a posição dos vértices.

Glossário

<code>attribute vec2 position;</code>	Struct que armazena um par ordenado
<code>gl_Position = vec4(position, 0.0, 1.0);</code>	Cria uma struct com 4 componentes (coordenadas homogêneas). Essa struct armazena a posição do fragmento (ponto).
<code>gl_FragColor = vec4(vermelho, verde, azul, alfa);</code>	Configura a cor de um fragmento (ponto).
<code>gl_PointSize</code>	Define o tamanho de um ponto (ponto flutuante).
<code>varying vec4 vColor</code>	Configura um valor de cor que pode mudar.
<code>on_init</code>	Função que configura a forma de renderização da tela.
<code>gl.glEnable(gl.GL_BLEND)</code>	Habilita renderização com blend de cores.
<code>gl.glBlendFunc(gl.GL_SRC_ALPHA, gl.GL_ONE_MINUS_SRC_ALPHA)</code>	Define os parâmetros do blend.

Exercícios:

- 1) Altere o último programa da prática 1 para exibir apenas um triângulo. Você pode escolher quais dos dois.
- 2) Crie uma função que recebe o click do mouse “on_mouse_press(x,y,button)” e atualiza a posição do triângulo.
- 3) Mude a função “on_mouse_press(x,y,button)” para identificar qual o vértice mais próximo do click. Adicione uma função “on_mouse_drag(x,y,dx,dy,button)”, quando o botão está pressionado mova a posição do vértice mais próxima.
- 4) Mude a função “on_mouse_press(x,y,button)” para mover o triângulo com o botão direito (button == 2) e alterar a posição do vértice com botão esquerdo (button == 1).