Texturas

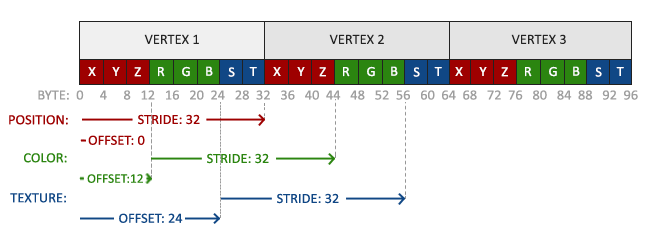
Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

|  |
| --- |
| using StbImageSharp; |

|  |
| --- |
| int texture;  texture = GL.GenTexture();  GL.BindTexture(TextureTarget.Texture2D, texture);  GL.TexParameter(TextureTarget.Texture2D, TextureParameterName.TextureWrapS, (int)TextureWrapMode.Repeat);  GL.TexParameter(TextureTarget.Texture2D, TextureParameterName.TextureWrapS, (int)TextureWrapMode.Repeat);  GL.TexParameter(TextureTarget.Texture2D, TextureParameterName.TextureMinFilter, (int)TextureMinFilter.LinearMipmapLinear);  GL.TexParameter(TextureTarget.Texture2D, TextureParameterName.TextureMagFilter, (int)TextureMagFilter.Linear);  string texturePath = "../../../src/Textures/container.jpg";  int width;  int height;  int nrChannels;  byte[] data;  StbImage.stbi\_set\_flip\_vertically\_on\_load(1);  ImageResult image = ImageResult.FromStream(File.OpenRead(texturePath), ColorComponents.RedGreenBlueAlpha);  width = image.Width;  height = image.Height;  //nrChannels = image.CompPerPixel;  data = image.Data;  if(data != null) {  GL.TexImage2D(TextureTarget.Texture2D, 0, PixelInternalFormat.Rgba, width, height, 0, PixelFormat.Rgba, PixelType.UnsignedByte, data);  GL.GenerateMipmap(GenerateMipmapTarget.Texture2D);  }  else {  Console.WriteLine("Failed to load texture");  } |

|  |
| --- |
| float[] vertices = {  // positions // colors // texture coords  -0.5f, -0.5f, 0.0f, 0.0f, 0.0f, 1.0f, 0.0f, 0.0f, // bottom left  -0.5f, 0.5f, 0.0f, 1.0f, 1.0f, 0.0f, 0.0f, 1.0f, // top left  0.5f, 0.5f, 0.0f, 1.0f, 0.0f, 0.0f, 1.0f, 1.0f, // top right  0.5f, -0.5f, 0.0f, 0.0f, 1.0f, 0.0f, 1.0f, 0.0f // bottom right  }; |



|  |
| --- |
| GL.VertexAttribPointer(0, 3, VertexAttribPointerType.Float, false, 8 \* sizeof(float), 0);  GL.EnableVertexAttribArray(0);  GL.VertexAttribPointer(1, 3, VertexAttribPointerType.Float, false, 8 \* sizeof(float), 3 \* sizeof(float));  GL.EnableVertexAttribArray(1);  GL.VertexAttribPointer(2, 2, VertexAttribPointerType.Float, false, 8 \* sizeof(float), 6 \* sizeof(float));  GL.EnableVertexAttribArray(2); |

shader.vert

|  |
| --- |
| #version 330 core  layout (location = 0) in vec3 aPos;  layout (location = 1) in vec3 aColor;  layout (location = 2) in vec2 aTexCoord;  out vec3 ourColor;  out vec2 TexCoord;  void main() {  gl\_Position = vec4(aPos, 1.0);  ourColor = aColor;  TexCoord = aTexCoord;  } |

shader.frag

|  |
| --- |
| #version 330 core  out vec4 FragColor;    in vec3 ourColor;  in vec2 TexCoord;  uniform sampler2D ourTexture;  void main() {  FragColor = texture(ourTexture, TexCoord);  } |

|  |
| --- |
| GL.BindTexture(TextureTarget.Texture2D, texture);  GL.BindVertexArray(VAO);  GL.DrawElements(PrimitiveType.Triangles, indices.Length, DrawElementsType.UnsignedInt, 0); |

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

|  |
| --- |
| FragColor = texture(ourTexture, TexCoord) \* vec4(ourColor, 1.0); |

Interface gráfica do usuário

Descrição gerada automaticamente