

INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Fundamentos de Computação Gráfica

CCI-36



Laboratório 4

COMP 20

Heládio Sampaio Lopes

Sebastião Beethoven Brandão Filho

Professor:

Carlos Henrique Q. Forster

ITA-SJC-2019



Descrição do código implementado

O Laboratório proposto possui como objetivo a criação de um shader em GLSL e utilização no THREE.JS.

Para o projeto do grupo utilizou-se um elemento de geometria esférica para representar o planeta Terra. Em seguida, foram utilizadas duas texturas distintas, uma representando a Terra de dia e outra representando a Terra à noite.

Utilizando as ferramentas associadas a um shader em GLSL realizou-se a transição entre essas duas texturas associadas, de forma a representar a passagem do dia por cada região da Terra, seguindo o movimento de rotação do planeta.

Para a realização do Laboratório foram seguidos determinados pré-requisitos estabelecidos. Para alguns tópicos são ilustrados determinados trechos de código implementando, objetivando a validação do pré-requisito associado.



1. Utilizar como material (fragment shader e/ou vertex shader) ou como efeito de imagem

O fragment shader foi usado para criar o efeito de transição entre duas imagens. O trecho de código abaixo ilustra tal elemento.

```
1 <script id="fragmentShader" type="x-shader/x-fragment">
2 uniform float time;
3 uniform vec2 resolution;
4 varying float intensity;
5 varying vec2 textureCoords;
6 uniform sampler2D mtexture;
7 uniform sampler2D ntexture;
8
9 void mainImage( out vec4 fragColor, in vec2 fragCoord )
10 {
11     vec2 xy = fragCoord.xy;
12
13     vec4 tx1 = texture2D(mtexture, xy);
14     vec4 tx0 = texture2D(ntexture, xy);
15
16     float ajust = 0.003;
17
18     float x = xy.x + ajust*time - floor(xy.x + ajust*time);
19
20     vec4 txt = vec4(abs(x-0.5)*tx0.r + (0.5-abs(x-0.5))*tx1.r,
21                     abs(x-0.5)*tx0.g + (0.5-abs(x-0.5))*tx1.g,
22                     abs(x-0.5)*tx0.b + (0.5-abs(x-0.5))*tx1.b,
23                     1.0);
24     fragColor = txt;
25 }
26 void main() {
27     vec4 myFragColor;
28     mainImage(myFragColor, textureCoords);
29     gl_FragColor = myFragColor;
30 }
31 </script>
```



2. Utilizar dados passados pela página em THREE.JS utilizando Uniforms

O uniforms foi utilizado para abarcar os elementos associados a textura do dia, textura da noite, resolução e tempo. Foi acrescentando, então, ao material da figura a ser criada.

```
1 uniforms = {
2   time: {type: "f", value: 1.0},
3   resolution: {type: "v2", value: new THREE.Vector2()},
4   mtexture: new THREE.Uniform(day),
5   ntexture: new THREE.Uniform(night)
6 };
7
8
9 material = new THREE.ShaderMaterial({
10   uniforms: uniforms,
11   vertexShader: document.getElementById('vertexShader').textContent,
12   fragmentShader: document.getElementById('fragmentShader').
      textContent
13 });
14
15 uniforms.resolution.value.x = window.innerWidth;
16 uniforms.resolution.value.y = window.innerHeight;
17
18 uniforms.time.value = 60. * elapsedSeconds;
```

Validação do código implementado

As Figuras 1 e 2 ilustram a passagem do tempo de uma determinada região do planeta de forma que inicialmente está iluminada e, em seguida, encontra-se escurecida.



Figura 1: Representação de uma região do planeta inicialmente iluminada.



Figura 2: Representação de uma região do planeta após escurecer.