

INSTITUTO TECNOLÓGICO DE AERONÁUTICA  
Fundamentos de Computação Gráfica  
CCI-36



**Laboratório 2**

**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 construção de uma geometria customizada, fazendo uso de um objeto Geometry ou de um objeto BufferGeometry, presentes na biblioteca THREE.JS. Para a sua realização, foram seguidos determinados pré-requisitos estabelecidos.

Para alguns tópicos são ilustrados determinados trechos de código implementando, que possui como objetivo a validação do pré-requisito associado.

### 1. Construir geometria, triângulo por triângulo, coordenada por coordenada

Para construir a geometria esférica proposta pelo grupo para o Laboratório foi necessário realizar um cálculo para o posicionamento de cada coordenada. Feito isso, os pontos definidos foram agrupados formando a geometria estabelecida. Na seção "POSICIONAR PONTOS" do código *esfera.js*, em anexo, tem-se a exemplificação da construção dessas geometrias.

### 2. Definir faces, coordenadas de textura e normais

Após a adição dos vértices, foi necessário saber em qual ordem eles deveriam ser inseridos para proporcionar a definição das faces. Dessa forma, associando pontos em grupos de 3 foi possível realizar a definição de todas as faces da geometria estabelecida.

### 3. Relatar como foram computados os índices das faces e garantir que a orientação das normais está correta

Para que as faces obtivessem uma orientação correta e condizente foi utilizada a "regra da mão direita", dessa forma, para que três pontos pudessem definir uma face, posicionou-se inicialmente algum deles em determinada posição e os outros foram sendo estabelecidos seguindo a orientação definida por essa regra a partir desse primeiro ponto. Na seção "DEFININDO FACES" do código *esfera.js*, em anexo, verifica-se a definição de faces na ordem correta para que as normais pudessem ficar "para fora" da geometria.

### 4. Inserir a geometria em um contexto "útil", por exemplo, criando uma animação em que varie a forma geométrica do objeto, ou suas cores ou suas coordenadas de textura.

De posse do sólido finalizado foi realizado o mapeamento da textura. Para isso, foi importante saber, também, em qual ordem as faces deveria ser definidas e qual seria a ordem dos pontos para cada face. Na seção "MAPEANDO TEXTURA" do código *esfera.js*, em anexo, verifica-se como foi mapeada a textura no entorno da geometria.

## Validação do código implementado

A ideia inicial para a geometria desenhada foi confeccionar uma esfera com dimensões pré-extabelecidas. Observou-se, porém, a possibilidade de escalar o processo, e assim foi implementada uma esfera com dimensões variáveis. Após desenhar a geometria foi inserida uma textura do Planeta Terra conforme apresenta a Figura 1.

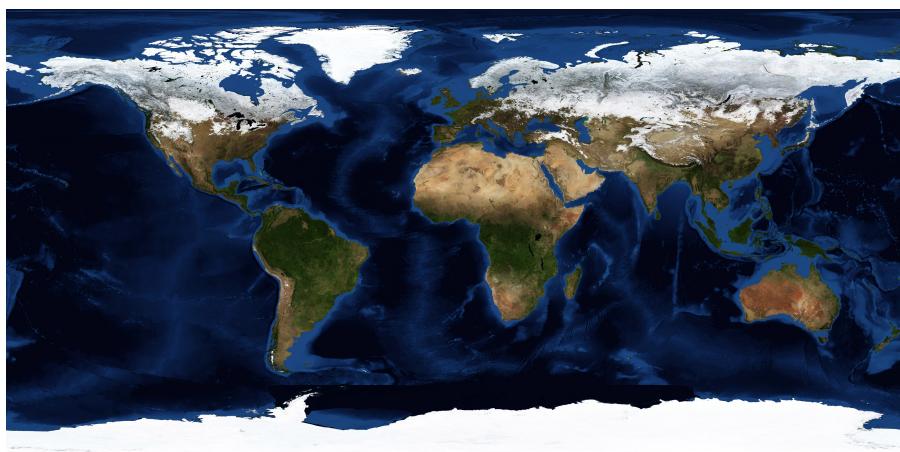


Figura 1: Figura representativa da Textura "Earth"utilizada.

As Figuras 2, 3 e 4, ilustram os resultados obtidos por meio da construção da geometria customizada proposta. A Figura 2 apresenta a geometria com dimensões "pequenas", ou seja, poucas camadas latitudinais e longitudinais. Já as Figuras 3 e 4 ilustram a geometria com dimensões maiores, demonstrando a proximidade com uma figura esférica de fato.

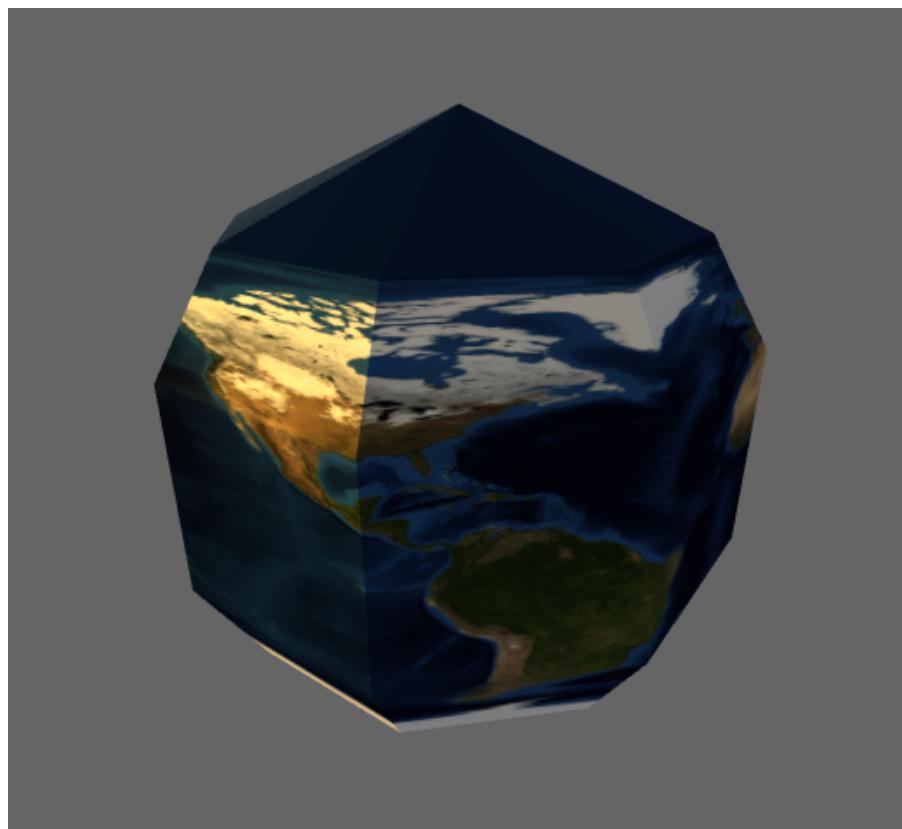


Figura 2: Figura representativa da Geometria Esférica associada à Textura "Earth".



Figura 3: Figura representativa da Geometria Esférica associada à Textura "Earth".



Figura 4: Figura representativa da Geometria Esférica associada à Textura "Earth".