



Universidade do Minho
Escola de Engenharia



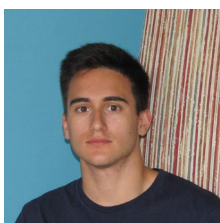
Universidade do Minho
Escola de Ciências

Computação Gráfica

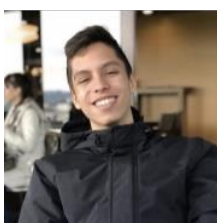
Guião 4

Grupo 16

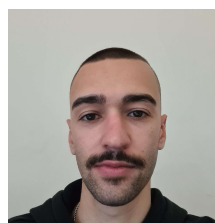
Carlos André Machado Costa a94543
João Miguel Rodrigues da Cunha a96386
Rúben Gonçalo Araújo da Silva a94633



a94543



a96386



a94633

27 de Maio de 2023

Índice

1	Introdução	2
2	Implementações	3
3	Generator	4
3.1	Normais/Indices de Texturas	4
3.1.1	Bezier	4
3.1.2	Torus	4
3.1.3	Sphere	4
3.1.4	Box	4
3.1.5	Plane	4
3.1.6	Cone	4
3.1.7	Cylinder	4
4	Engine	5
4.1	Material	5
4.2	Lights	5
4.3	Textures	5
4.4	Sistema Solar	6
5	Demos	7
6	Conclusão	9
7	Bibliografia/Webgrafia	10

1 Introdução

Durante a quarta fase deste projeto, no âmbito da disciplina de Computação Gráfica, foi proposta a implementação de um sistema de luzes e texturas. Nesta nova etapa, o nosso grande objetivo era tornar o sistema solar num ambiente mais realista, adicionando efeitos de iluminação e aplicar texturas nos nossos objetos.

2 Implementações

De momento foi implementada as seguintes funcionalidades e scenes:

- Generator
 - Calcular as Normais
 - Calcular os indices para as texturas
- Engine:
 - Novas Features:
 1. Sistema de Luzes;
 2. Texturas.
- Scenes:
 - Finalizar Sistema Solar.

3 Generator

3.1 Normais/Indices de Texturas

3.1.1 Bezier

As normais no Bézier foram calculadas com a fórmula do cross product. Basicamente, a cada três vértices obtidos (formando um triângulo), foi calculado o cross product e normalizado. Esse vetor é a normal desses três vértices. Para calcular os índices de texturas, seguimos a tesselação sem realizar nenhum cálculo, visto que são compreendidos entre 0 e 1.

3.1.2 Torus

No toro, as normais foram calculadas simplesmente dividindo o raio de cada componente de cada vértice. Os índices foram obtidos ao dividir 1 pela quantidade *sides* e pela quantidade de *rings*.

3.1.3 Sphere

Na esfera as normais foram calculadas simplesmente dividindo o raio a cada componente de cada vértice. Os índices foram obtidos a dividir 1 pela quantidade de *stacks* e pela quantidade de *slices*.

3.1.4 Box

Na box as normais foram atribuídas considerando a face em questão, onde todos os vértices da mesma face compartilham da mesma normal. Os índices foram obtidos dividindo 1 pela quantidade de divisões em ambas as componentes.

3.1.5 Plane

No plano, as normais foram atribuídas todas com $z=1$. Os índices foram obtidos simplesmente dividindo 1 pela quantidade de divisões em ambas as componentes.

3.1.6 Cone

No cone, as normais foram atribuídas com $z=-1$ na base, e na lateral foi necessário calcular a inclinação do cone. Para isso calculamos a razão entre o raio e a altura (*Radius/Height*), depois aplicamos a função $\text{atan}()$ para obter o ângulo e, por fim, aplicamos o $\sin()$ para obter a altura em radianos. Os índices foram calculados com base na quantidade de *stacks* e *slices* na *lateral*. Na base, deslocamos o cálculo para o centro da imagem hipotética, dessa forma, ao obtermos índices maiores ou menores que $[0,1]$, obtemos uma textura simétrica, visto que a textura se repete infinitamente.

3.1.7 Cylinder

No cilindro, as normais foram atribuídas todas para $z=-1$ na base inferior e $z=1$ para a base de cima, e na lateral seguimos a trigonometria básica da base do cilindro. Os índices foram calculados seguindo a quantidade de *stacks* e *slices*, isto na lateral. Na base deslocamos o cálculo para o centro da imagem hipotética e com isso, ao obtermos índices maiores ou menores que $[0,1]$ obtemos uma textura simétrica visto que a textura se repete infinitamente.

4 Engine

Nesta fase, a nossa engine consegue definir os materiais de cada objeto, especificar onde e que tipos de luz queremos (até 8) e colocar texturas se for pedido.

4.1 Material

Cada objeto tem as seguintes propriedades de material:

1. diffuse;
2. ambient;
3. specular;
4. emissive;
5. shininess.

4.2 Lights

Cada light tem uma definição base de:

1. ambient: [0.2f, 0.2f, 0.2f, 1.0f];
2. diffuse: [1.0f, 1.0f, 1.0f, 1.0f];
3. specular: [1.0f, 1.0f, 1.0f, 1.0f]

Também temos a luz ambient com as seguintes definições: [0.3f, 0.3f, 0.3f, 1.0f]

Para que a lightning faça sentido nos nossos objetos, é necessário, em primeiro lugar, ter as normais de cada vértice e implementá-las previamente através de VBOs (Vertex Buffer Objects).

4.3 Textures

Para as texturas façam sentido nos nossos objetos, é necessário em primeiro lugar, termos os índices de cada vértice e estes serem previamente implementadas por VBOs. Também foi usada a função dada nas aulas práticas para obtermos o processamento das texturas e estas serem guardadas no sistema.

4.4 Sistema Solar

Nesta secção, temos alguns frames do nosso sistema solar. Para melhor visualização é recomendado assistir o vídeo gravado.



Figura 1: solar_system_1

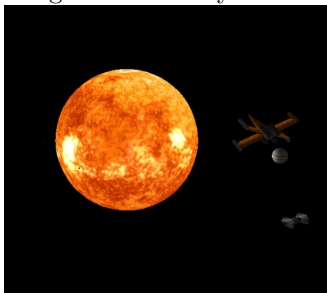


Figura 3: solar_system_3

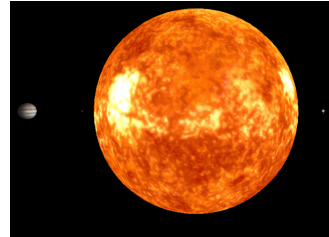


Figura 2: solar_system_2

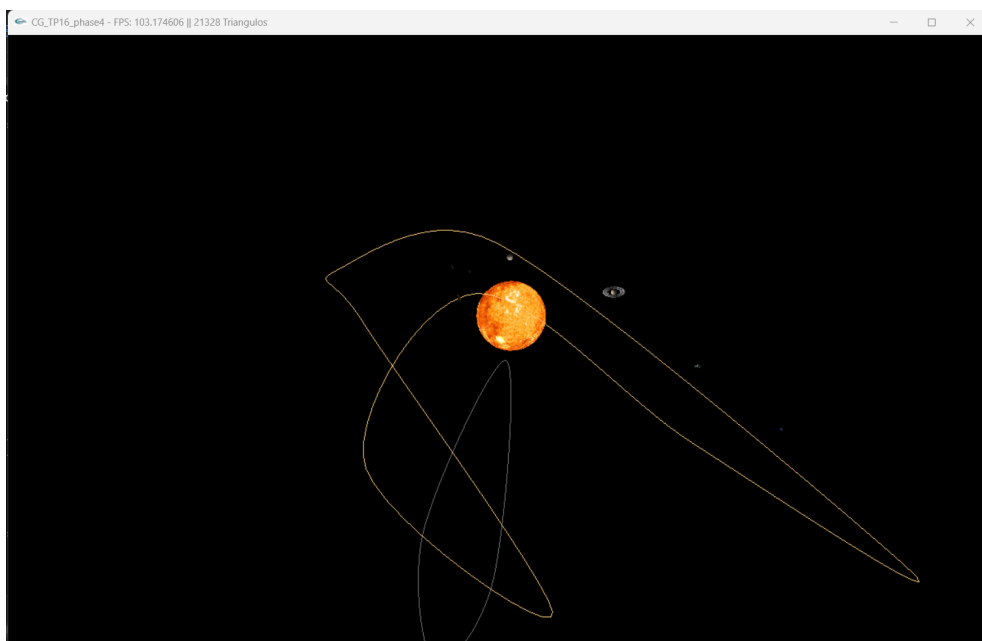


Figura 4: solar_system_4

5 Demos

Nesta secção, apresentamos os testes pretendidos. Para demonstrar melhor o sistema solar, foi gravado um vídeo a mostrar todas as capacidades da engine e o sistema solar em si. Este arquivo encontra-se nos ficheiros entregue.

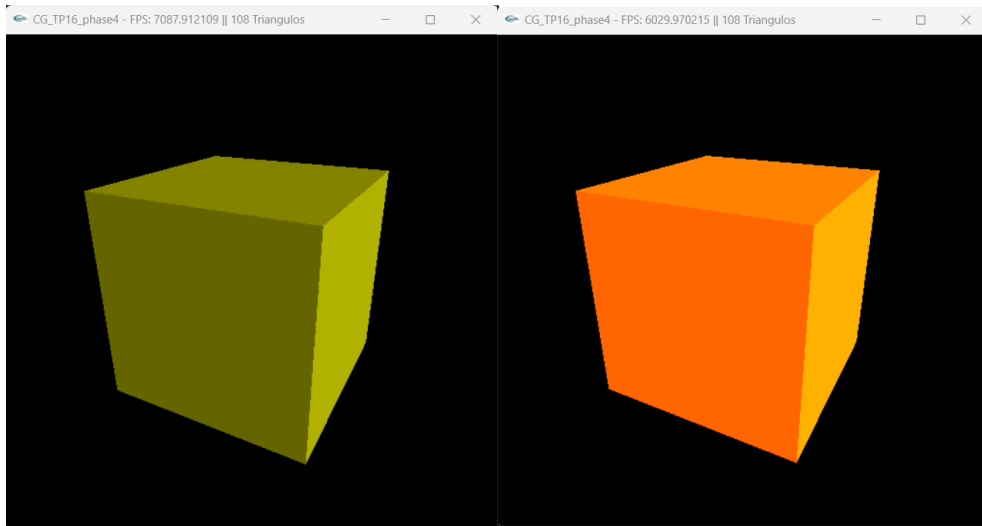


Figura 5: test_3.1

Figura 6: test_3.2

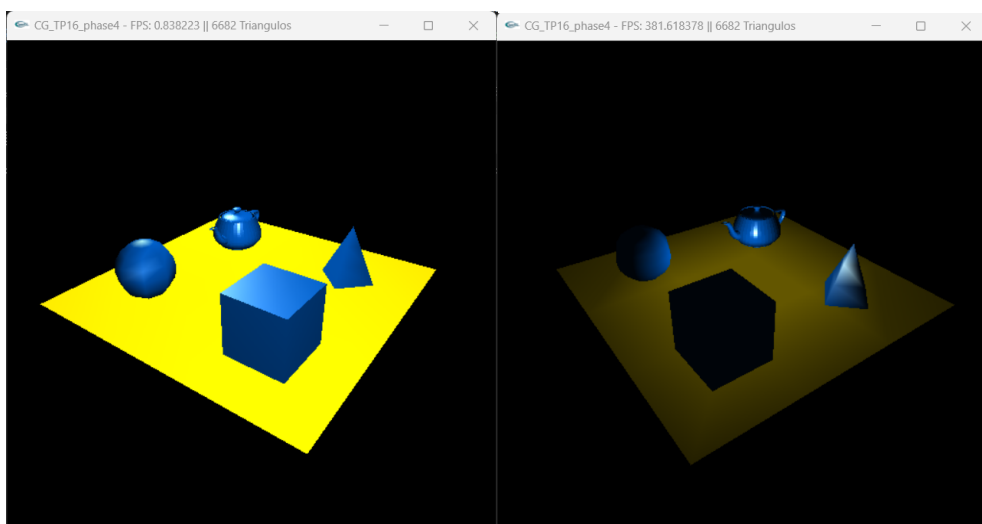


Figura 7: test_3.1

Figura 8: test_3.2

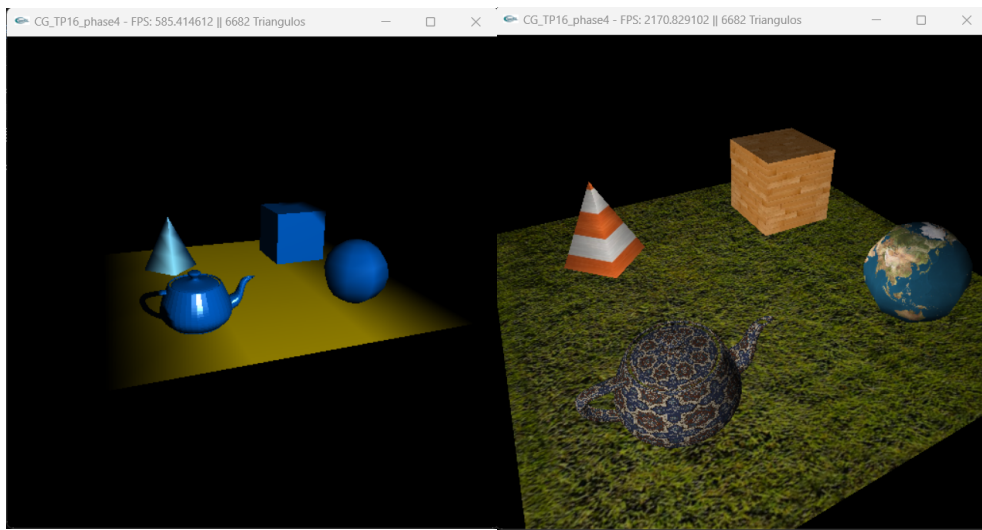


Figura 9: test_3.1

Figura 10: test_3.2

6 Conclusão

Durante esta quarta fase tivemos de lidar com situações difíceis como o cálculo das normais de cada objeto e a determinação dos seus índices para uma possível textura. Também enfrentamos um desafio na implementação dos materiais na engine e as luzes. Nesta fase em particular, sentimos que foi a mais importante pois foi necessário que tudo fosse criado na perfeição. Qualquer erro mínimo poderia comprometer como o objeto seria representado.

7 Bibliografia/Webgrafia

- Material da cadeira de Computação Gráfica 22/23