

Universidade do Minho

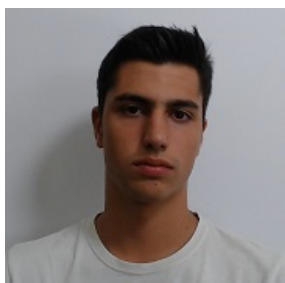
Computação Gráfica

Fase 4 - Normais e Coordenadas de Textura

Grupo 31

Mestrado Integrado em Engenharia Informática

A89466



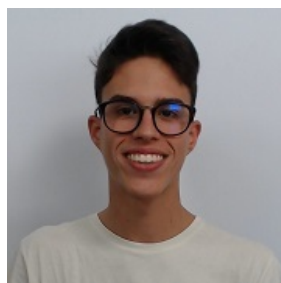
João Amorim

A89511



Rodrigo Meira

A89550



João Sá

Maio de 2020

Conteúdo

1	Introdução	3
2	Estrutura do Projeto	3
3	Iluminação	4
3.1	Plano	4
3.2	Box	4
3.3	Cone	4
3.4	Sphere	4
4	Texturas	6
5	Parser	6
6	Testes	6
7	Conclusão	8

1 Introdução

Para a unidade curricular de Computação Gráfica foi nos proposto criar uma mini engine capaz de desenhar cenas com desenhos 3D que se encontra dividido em 4 fases. Como 4^a e última fase do projeto, foi-nos pedido para adicionar iluminação e texturas.

2 Estrutura do Projeto

A estrutura do projeto não foi alterada para esta fase, pelo que continuamos com uma engine e um generator.

Testado em: Esta fase do trabalho prático de computação gráfica foi testada somente no sistema operativo windows.

Nota: A pasta toolkits é necessária para a engine, mas não está contida no projeto uma vez que foram os docentes que a disponibilizaram.

Como biblioteca externa foi usado o *tinyxml2* (1) para *parsing* dos ficheiros xml.

3 Iluminação

Para colocar iluminação no nosso projeto foi necessário fazer alterações tanto na engine como no generator. No generator foi necessário alterar a maneira de escrever os ficheiros .3d, pois para a iluminação também precisamos das normais de cada vértice. A solução do nosso grupo foi calcular as normais e escrevê-las no ficheiro .3d por baixo dos índices dos vértices. Como na fase anterior depois dos vértices só havia índices, acrescentamos também uma linha que indica o número de índices de vértices que estão escritos e precisam de ser lidos, esta linha foi acrescentada entre o último vértice escrito e o primeiro índice escrito. Em relação às normais estas foram definidas mediante a figura geométrica que queríamos gerar.

3.1 Plano

As normais do plano foram as mais simples de fazer, pois como o plano é visto de cima as normais vão ser todas iguais para cada vértice, sendo cada normal definida como $(0,1,0)$.

3.2 Box

Para a caixa utilizamos normais diferentes para cada face, na face virada para cima utilizamos normais iguais às do plano, na face virada para baixo utilizamos as normais simétricas à face de cima. Para as faces laterais utilizamos apenas normais horizontais, ou seja, dependendo da face lateral só alteravamos a coordenada do X e/ou do Z. Para os vértices da face lateral de frente para a câmara e do lado direito definimos as normais $(1,0,0)$, para a face lateral de frente para a câmara e do lado esquerdo definimos as normais $(0,0,1)$, as faces opostas têm as normais simétricas.

3.3 Cone

Em relação ao cone definimos as normais da base, que são iguais às da face de baixo da caixa, definimos as normais dos vértices do cone como $(0,1,0)$, para calcular as normais das faces laterais utilizamos o exemplo do cilindro das aulas práticas, definindo como $(\cos(\alpha*i), 0, \sin(\alpha*i))$, sendo a variável i corresponde à slice que estamos a desenhar.

3.4 Sphere

No que diz respeito às normais da esfera, para as calcular também recorremos aos slides das aulas práticas e apenas normalizamos as coordenadas dos vértices que desenham os triângulos da esfera.

Nas imagens representadas em baixo temos as implementações dos métodos de iluminação de objetos, das figuras pedidas na primeira fase que são, respetivamente o plano, a caixa, o cone e, por fim, a esfera.

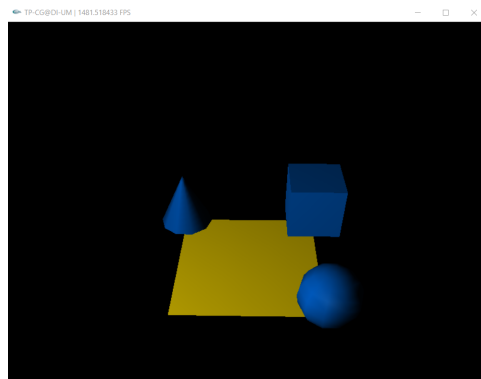


Figura 1: Imagem de Iluminação

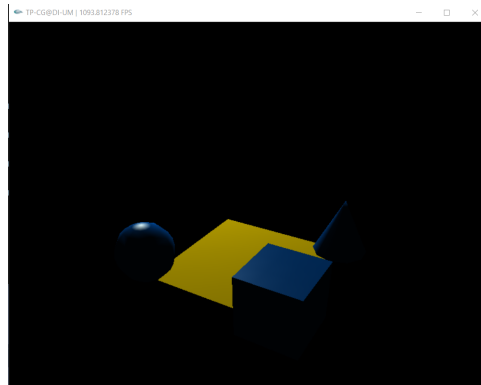


Figura 2: Imagem de Iluminação

4 Texturas

Para esta última fase do trabalho prático implementamos o componente da texturização para quase todas as figuras geométricas propostas na primeira fase, somente não as fizemos para o cone.

Utilizamos também as imagens fornecidas, pelo corpo docente, para colocar nas esferas, de forma a representar os respectivos planetas. Tal como, está demonstrado num dos ficheiros de teste enviados.

5 Parser

Para esta última fase foi necessário acrescentar novos campos denominados como `lights` e `color`.

O primeiro é, relativo à fonte de luz e as suas coordenadas, dado que as mesmas tanto podem ser direção como posição conforme o tipo da fonte de luz que estamos a ler. Por fim, o `color` (filho do `model`) apresenta as diferentes definições de luz e as coordenadas RGB das mesmas.

Para finalizar, adicionamos também uma nova linha dentro do `model` que enuncia o nome do ficheiro da textura.

6 Testes

Na seguinte imagem, temos um exemplo de um teste feito pelo nosso grupo de trabalho.

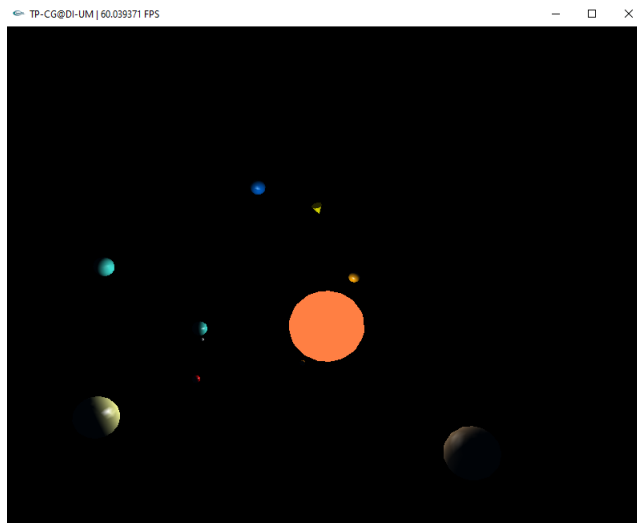


Figura 3: Sistema Solar com iluminação



Figura 4: Sistema Solar com texturas

7 Conclusão

Nesta última fase do trabalho prático da unidade curricular de Computação Gráfica fomos capazes de implementar os métodos de iluminação de objetos como foi demonstrado e explicado anteriormente e a texturização. Como grupo sentimos dificuldades em gerir o tempo, nomeadamente na fase 3, que ficou muito incompleta, e nas restantes fases para conseguir resolver certos bugs e adicionar funcionalidades extra. Para terminar, à medida que as fases do trabalho, propostas pelo corpo docente, foram concebidas e desenvolvidas, aprofundamos os nossos conhecimentos, relativamente a esta unidade curricular.

Referências

[1] tinyxml2 - <https://github.com/leethomason/tinyxml2>.