

Universidade do Minho
Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Computação Gráfica

Relatório do Trabalho Prático 4

A78890 Alexandre Reis da Costa
A75248 Ana Sofia Gomes Marques
A65277 Flávio Manuel Machado Martins
A79799 Gonçalo Nogueira Costeira

Grupo 50

31 Maio 2020

Conteúdo

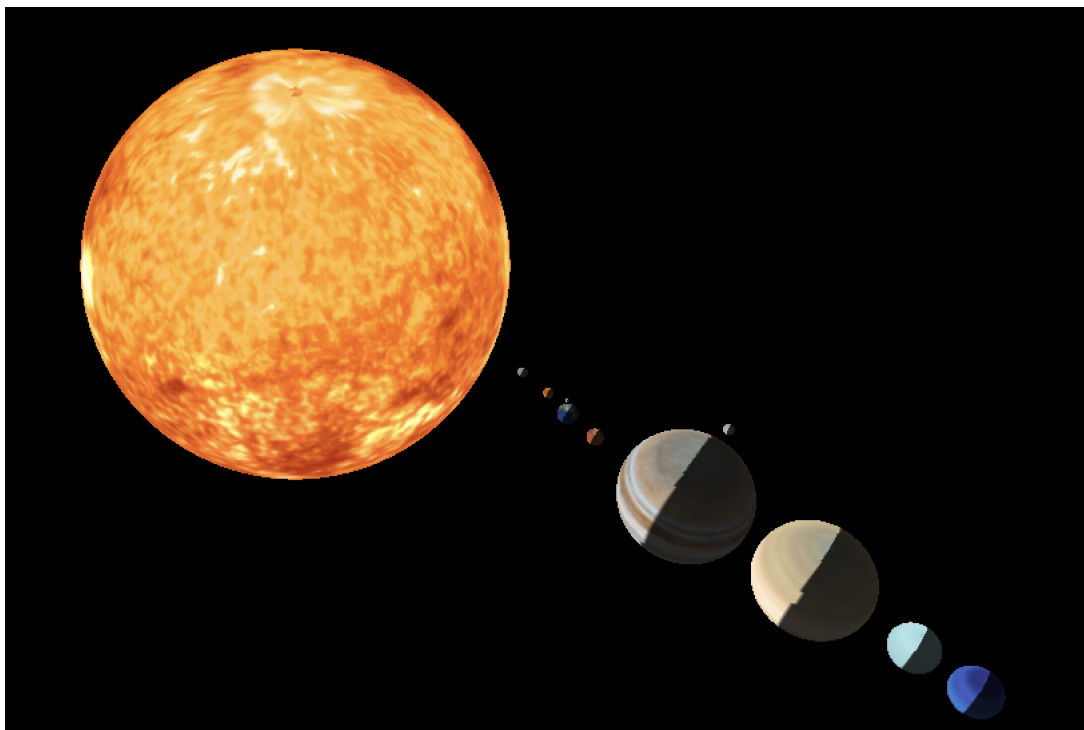
1	Introdução	i
2	Generator	ii
3	Luz	iii
4	Textura	iv
5	Engine	v
6	Sistema Solar	vi
6.1	input.xml	vi
7	Conclusão	vii
8	Referências	viii

1 Introdução

O presente documento diz respeito à quarta fase do projeto proposto no âmbito da unidade curricular de Computação Gráfica.

Esta etapa tinha como principais objetivos, a aplicação de primitivas para simular os efeitos da luz e a utilização de texturas para conferir realismo ao modelo.

Neste relatório descreve-se detalhadamente cada uma das componentes da quarta e última fase. Começa-se por descrever as novas funcionalidades do novo programa, bem como o modo de utilização do mesmo.



2 Generator

As atualizações começaram por ser feitas nesta classe. Para gerar os vetores necessários para implementar a luz e as texturas foi necessário alterar o processo que gera as figuras geométricas.

As normais dos pontos de um objecto servem para que o OpenGL possa calcular o comportamento da luz ao longo das superfícies da primitiva.

As coordenadas de textura servem para que se possa atribuir uma textura a uma primitiva. Cada ponto terá um par de coordenadas x e y, que variam entre 0 e 1.

```
void sphere(float radius, int slices, int stacks, FILE *f){
    ....
    ...vector<float> vertices;
    ...vector<float> normais ;
    ...vector<float> textura ;
```

```
void box ( int ndivs, float xx, float yy, float zz, FILE *f ){
    ....
    ...vector<float> vertices;
    ...vector<float> normais ;
    ...vector<float> textura ;
```

```
void cone(float radius, float altura, int slices, int stacks, FILE *f) {
    ....
    ...vector<float> vertices;
    ...vector<float> normais ;
    ...vector<float> textura ;
```

```
void plano(float tam, FILE *f){
    ....
    ...vector<float> vertices;
    ...vector<float> normais ;
    ...vector<float> textura ;
```

3 Luz

Foram definidos três tipos de luzes:

-Luz pontual: Os objetos que possuem este tipo de comportamento, definido na classe `SpotLight`, emitem luz em todas as direções.

-Luz direcional: Este modelo, definido na classe `directionLight`, parte do princípio que a fonte de luz está infinitamente longe do alvo e por isso os feixes de luz são todos paralelos.

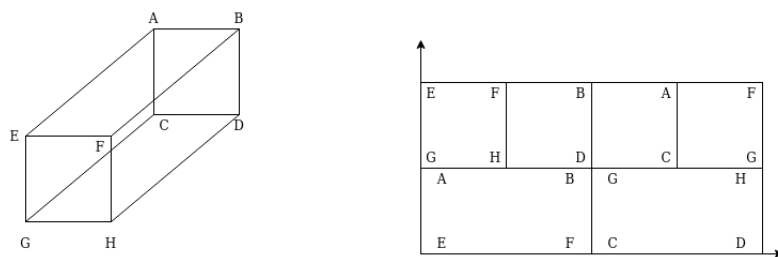
-Luz spotlight: Este tipo de luz, definida na classe `SpotLight`, pretende recriar um holofote virtual e como tal é possível configurar não só a sua localização como também a direção para a qual ele aponta.

4 Textura

As coordenadas definidas no array textura são utilizadas pelo OpenGL para saber que ponto na imagem a ser utilizada como textura corresponde ao ponto da primitiva em questão.

Plano: Plano: neste caso basta mapear as extremidades do plano para a extremidade correspondente da imagem, isto é , canto superior direito do plano para (1,1), canto superior esquerdo do plano para (0,1), canto inferior direito do plano para (1,0) e o canto inferior esquerdo do plano para (0,0) .

Caixa: Para o caso da caixa, utilizando como textura um ficheiro dividido em 2 partes de igual, a parte superior contendo dividida em 4 retangulos iguais correspondentes aos 4 lados, e a metade inferior contendo a parte superior e inferior da caixa(tambem com uma divisão a metade).



Mapeamento de textura numa caixa.

Esfera: sendo alpha e beta os ângulos que variam, respectivamente $[0..2*PI]$ e $[-PI/2 \dots PI/2]$, facilmente se mapeiam para espaço textura (u e v ambos a variar entre 0 e 1) com a transformação $u = a / (2 * PI)$ e $v = (b / PI) + 0.5$.

5 Engine

Nesta fase do projeto decidimos remover a componente de cor que tínhamos nas fases anteriores, uma vez que esta não se mostrou necessária uma vez que ao carregar a textura esta se sobrepunha á cor já atribuída. A cor foi então removida de todos os ficheiros em que se encontrava implementada, bem como no input.xml.

6 Sistema Solar

6.1 input.xml

No input.xml, ficheiro xml passado como parâmetro na função main presente no sistemSolar.cpp, foi adicionado um ponto de luz na posição (0,10,0), criando o efeito pretendido de um sistema Solar mais próximo da realidade, mas também existe a possibilidade de ser adicionado um foco de luz e/ou uma luz direcional.

No sub-grupo models foi adicionada a textura a todos os planetas, respectivas luas e ainda ao Sol, foi ainda neste grupo implementadas luzes, sendo no caso do sol emissiva, pois é o sol que imite a luz que vai iluminar os outros astros, ambiente e difusa, significa que a luz segue em todas as direções. Nos planetas e luas é implementada luz ambiente, difusa (como no sol) e especular, que garante que a luz que recebem pelo Sol é refletida.

Foi ainda feito um ajuste face ao tamanho dos astros e o seu espaçamento por forma a ser mais fácil visualizar as novas implementações.

7 Conclusão

O desenvolvimento desta quarta e última fase do projeto, demonstrou que a geometria e cálculos matemáticos mal efetuados têm um profundo impacto no esquema final, que faz desta a fase mais complexa do projeto. A complexidade por detrás das luzes e texturas aplicadas foi o grande obstáculo desta fase, uma vez que levou a alterações à forma como as figuras geométricas são geradas e posteriormente tratadas.

Esta fase permitiu "desmistificar" o processo que permite simular o efeito da luz e o processo de replicação da aparência real dos objetos a renderizar.

Como trabalho futuro consideramos que o cometa implementado na terceira fase poderá ser modificado para funcionar de acordo com as novas primitivas.

De modo geral, concluímos que este projeto foi efectuado com sucesso e estamos bastante contentes com o resultado final do nosso sistema solar.

8 Referências

Lighthouse3d. (2017). GLUT Tutorial. online Available at:
<http://www.lighthouse3d.com/tutorials/glut-tutorial/>

Opengl-tutorial.org. (2017). Tutorial 3 : Matrices. online Available at:
<http://www.opengl-tutorial.org/beginners-tutorials/tutorial-3-matrices/>