

Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Computação Gráfica

Ano Letivo de 2022/2023

Fase 4 – Normais e Coordenadas de Texturas

Grupo 6

A96434 Francisca Quintas Monteiro de Barros
A97588 Joana Isabel Freitas Pereira
A94870 Rafael Picão Ferreira Correia

4 de junho de 2023

Índice

Lista de Figuras	3
1 Introdução	4
2 Generator	5
2.1 Figuras Extra	5
2.1.1 Cilindro	5
2.1.2 Anel	6
2.1.3 Caixa Invertida	7
3 Engine	8
3.1 Cores	8
3.2 Lighting	8
3.3 Texturas	8
3.4 Extras	8
3.4.1 IBOs	8
3.4.2 Suporte de ficheiros .obj	9
3.4.3 Controlar a Câmera	9
3.4.4 Funcionalidades extra no XML	10
4 Testes	11
4.1 Teste 1	11
4.2 Teste 2	12
4.3 Teste 3	12
4.4 Teste 4	13
4.5 Teste 5	13
4.6 Teste 6	14
5 Demo Scenes	15
5.1 Sistema Solar	15
5.1.1 Alterações do sistema solar	15
5.2 Torre com Cavaleiro	16
5.3 Cidade	16
6 Conclusão	17

Lista de Figuras

2.1	Cilindro	6
2.2	Anel com raio interior 2 e exterior 4	6
2.3	Textura da caixa invertida	7
2.4	Caixa invertida com tamanho 100	7
4.1	Teste 1	11
4.2	Teste 2	12
4.3	Teste 3	12
4.4	Teste 4	13
4.5	Teste 5	13
4.6	Teste 6	14
5.1	Sistema Solar	15
5.2	Torre	16

1 Introdução

O presente relatório é referente à quarta e última fase do trabalho prático desenvolvido pelo grupo 06 no âmbito da Unidade Curricular de Computação Gráfica, lecionada no curso de Licenciatura em Engenharia Informática no 2º Semestre do ano letivo 2022/2023.

Este projeto tem como objetivo o desenvolvimento de um mecanismo 3D baseado num cenário gráfico, através de ferramentas como o OpenGL e o C++, também usadas nas aulas práticas.

O projeto está dividido em 4 fases e para cada fase, serão fornecidos arquivos de configuração XML para testes e avaliação.

Nesta última fase o grupo alterou completamente o funcionamento tanto na *Engine* como no *Generator* de modo a suportar novas funcionalidades. Na *Engine*, foi substituído o método de desenho *Vertex Buffer Objects* (VBOs) para *Index Buffer Objects*, o que resultou num melhor desempenho e maior eficiência na renderização dos modelos. Além disso, o grupo implementou cores, texturas e luzes.

No *Generator*, foi implementado normais, texturas e índices em cada um dos objetos já existentes. Além dos elementos essenciais para esta fase, também foram incluídos uma variedade de outros objetos, como uma caixa invertida, um anel e um cilindro.

Por fim, as melhorias foram implementadas no sistema solar, adicionando os novos recursos de texturas e iluminação aos planetas e cometa. Essas adições aumentaram o realismo e a interatividade da simulação.

2 Generator

Nesta fase, foi necessário fazer alterações profundas ao *Generator* de modo a este poder suportar IBOs, iluminação e texturas. Além disso, a fim de completar e tornar mais realista o sistema solar e as *demoscenes* criadas pelo grupo foram adicionadas figuras extras.

Para que o *Engine* processe corretamente a iluminação com base na luz e no objeto, foi necessário, para cada ponto, definir a sua normal.

No que diz respeito às texturas, foi imprescindível atribuir a cada ponto as coordenadas de textura apropriadas. Essas coordenadas correspondem a um mapeamento entre as informações da textura e as superfícies dos objetos. Isso permite uma representação visual mais realista e coerente.

No caso das coordenadas de textura da base do cone, surgiram algumas dúvidas, uma vez que a textura fornecida era um retângulo. O grupo optou por assumir que a textura nos pontos da base teriam a cor correspondente ao ponto de textura para esse mesmo ponto mas nas faces do cone, ou seja, no caso da textura dada, a base seria toda laranja.

Em relação aos IBOs (Index Buffer Objects), foram removidos os pontos repetidos que estavam a ser guardados e passou-se a indicar o índice do ponto a desenhar. Isso otimizou o uso de memória e melhorou a eficiência do processo.

Assim, a estrutura de um ficheiro criado pelo *Generator* passa a ter a seguinte estrutura: na primeira linha um lista de coordenadas separadas por espaço, que representam os pontos, as normais e as coordenadas de textura - para cada ponto tem 3 coordenadas de posição, 3 coordenadas das normais e 2 coordenadas da textura seguidas. Na linha seguinte, a lista dos índices.

Esta foi a estratégia mais simples que se encontrou visto que não é necessário saber o número de pontos, é só ir lendo a linha, pois sabe-se que tudo corresponde a pontos (lê-se 8 coordenadas de uma vez, que correspondem a um ponto). Depois a linha seguinte são apenas índices.

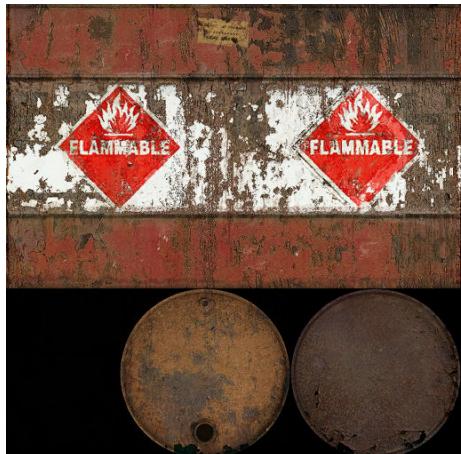
2.1 Figuras Extra

2.1.1 Cilindro

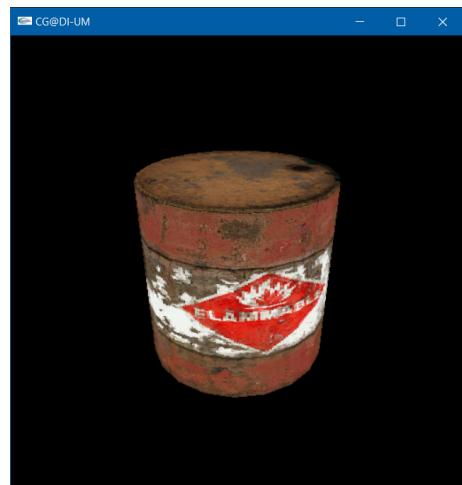
A criação de um cilindro requer a definição de três parâmetros importantes: altura, raio e número de lados.

O cilindro é centrado na origem e é desenhado por fatias e, para cada fatia é seguida a seguinte ordem: a face de cima, a base e os dois triângulos laterais.

As texturas para este modelo devem ser adequadas e cobrir toda a figura, tal como a textura usada na aula:



(a) Textura cilindro



(b) Cilindro com 1 raio e 2 de altura

Figura 2.1: Cilindro

2.1.2 Anel

Para o sistema solar, surgiu a necessidade de criar um anel. Este anel é bidimensional, visto que se assemelha com um plano, mas circular e delimitado por dois círculos, um mais pequeno e um maior. Para tal, é necessário passar 4 parâmetros: o raio do círculo interior, o raio do círculo exterior, o número de lados e o número de camadas.

Foi decidido, no anel, ter as duas faces visíveis, visto que, no sistema solar este estará inclinado, no entanto, assim é necessário desenhar o dobro dos triângulos.

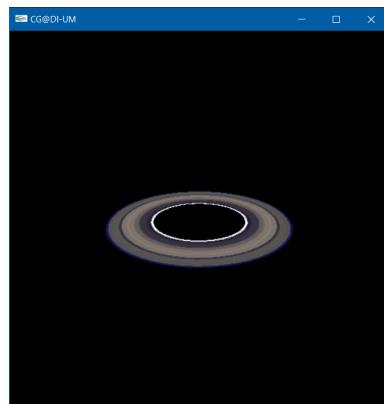


Figura 2.2: Anel com raio interior 2 e exterior 4

2.1.3 Caixa Invertida

Tal como o anel, para o sistema solar surgiu a ideia de colocar um fundo de estrelas, e a estratégia pensada foi criar uma caixa invertida com uma textura de estrelas. Para esta caixa, foi seguida uma estratégia idêntica à caixa normal, no entanto, os triângulos foram desenhados "ao contrário", com a face para dentro. No entanto, não se obteve o resultado esperado visto que a textura se repetia em cada face, havendo cortes notáveis. Assim, foi feita uma nova versão (foi deixada a mais antiga e acrescentada uma nova). Esta nova versão usa texturas deste tipo:

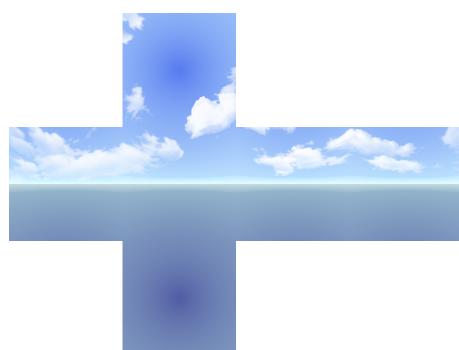
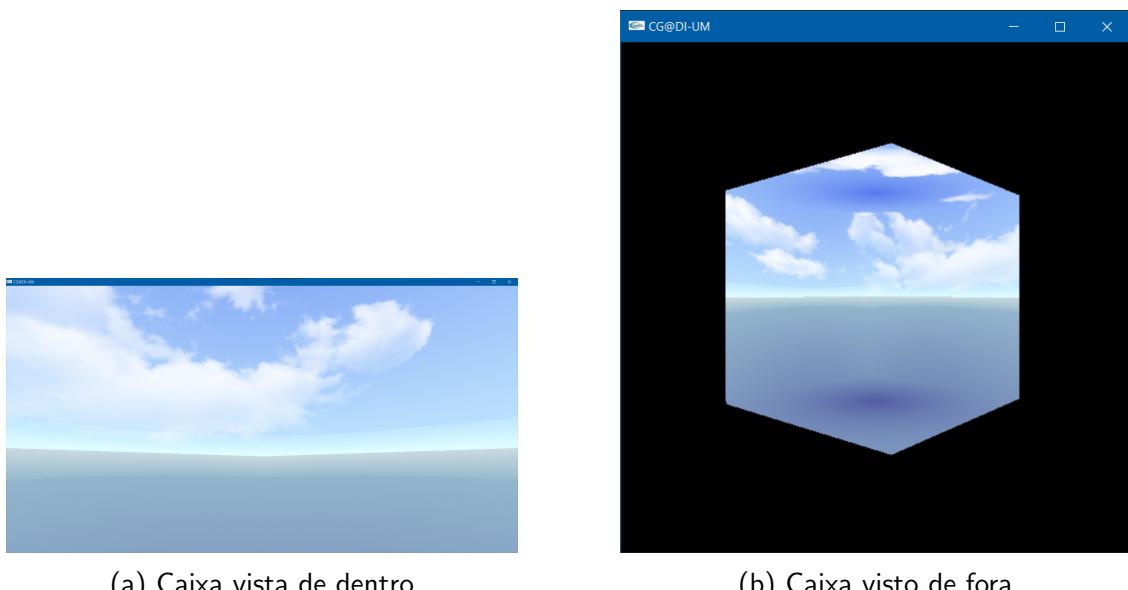


Figura 2.3: Textura da caixa invertida



(a) Caixa vista de dentro

(b) Caixa visto de fora

Figura 2.4: Caixa invertida com tamanho 100

3 Engine

3.1 Cores

Inicialmente surgiram algumas dificuldades, uma vez que as cores dos objetos estavam a afetar a cor dos eixos. Para resolver esse problema foi desativada a cor e as texturas ao desenhar os eixos e desenhou-se os eixos em modo imediato. Usou-se glColor para "pintar" os eixos.

3.2 Lighting

As luzes foram implementadas com sucesso. Assumiu-se que no máximo há 8 luzes e sempre que se cria uma nova luz ativa-se a respetiva luz e muda-se a cor da mesma para branco (já que por defeito as luzes 2-8 são pretas).

3.3 Texturas

As texturas foram implementadas com sucesso e optou-se por usar Mipmapping de modo a melhorar a qualidade visual.

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST_MIPMAP_NEAREST);
```

3.4 Extras

3.4.1 IBOs

Nesta fase foi necessário refazer a estrutura do trabalho para suportar IBOs, em vez de ter tudo no engine e guardar todos os pontos de todos os models no buffer, optou-se por criar um

objeto para cada model e ter um buffer para cada model em vez de ter 1 global. Isto porque na função

```
glDrawElements(GL_TRIANGLES, this->indexesCount, GL_UNSIGNED_INT, 0);
```

O ultimo parâmetro que supostamente é o *offset*, ou seja aonde devíamos começar a escrever no *buffer*, está *deprecated*. Para isso, tivemos que ter um *buffer* para cada *model* que carregávamos, em vez de ter um *buffer* geral aonde metíamos todos os pontos. Assim o nosso *engine* passou a funcionar com *IBOs*, seguindo um paradigma de objetos.

3.4.2 Suporte de ficheiros .obj

A partir do uso da biblioteca tiny object loader foi possível implementar o suporte de ficheiros *.obj* seguindo a documentação da biblioteca usada localizada no *github*. Para cada ficheiro *obj* nós ignorámos os materiais que estão no ficheiro e só retirámos os pontos, as normais e as texturas. Sempre que desenhamos um *obj* usamos um VBO porque os ficheiros contêm diferentes indices para as texturas, normais e pontos, o que não seguia a estrutura de como implementamos os IBOs.

3.4.3 Controlar a Câmera

Para controlar a câmera, o grupo decidiu implementar dois modos distintos. No modo 1, a câmera é fixada num ponto específico e é possível alterar a posição da mesma para percorrer uma esfera ao redor desse ponto para onde está a olhar. Essa abordagem permite visualizar a cena de diferentes ângulos, mantendo o foco no ponto de interesse.

No modo 2, a câmera tem uma liberdade de movimento mais simples e direta. Ela pode mover-se para a frente, para trás, para a direita, para a esquerda, para cima e para baixo.

Para alternar entre os dois modos, é necessário clicar na tecla 'r'. Isso permite que os utilizadores escolham a abordagem de controlo mais adequada às suas necessidades e preferências, oferecendo flexibilidade na interação com a câmera.

Aqui segue-se a lista dos controlos:

Tecla	Modo	Descrição
T	1/2	Mostar/Esconder os Eixos
G	1/2	Mostar/Esconder as Linhas criadas pelos translates ao longo do tempo.
H	1/2	Mudar o modo Polygon (GL_FILL, GL_POINT, GL_LINE)
R	1/2	Mudar de Modo
W	1	Vai para cima, mas continua a olhar para o mesmo sitio
S	1	A câmera vai para baixo, mas continua a olhar para o mesmo sitio
A	1	A câmera vai para a esquerda, mas continua a olhar para o mesmo sitio
D	1	A câmera vai para a direita, mas continua a olhar para o mesmo sitio
Z	1	Zoom
X	1	Unzoom
W	2	Mover em frente
S	2	Mover para trás
A	2	Mover para a esquerda
D	2	Mover para a direita
V	2	Mover para cima
C	2	Mover para baixo

3.4.4 Funcionalidades extra no XML

Com as melhorias feitas na *Engine* é necessário especificar as novas funcionalidades no xml.

No elemento *world*, agora existe a funcionalidade *axisenable*, que pode ser configurada como *true* para ativar os eixos ou *false* para desativá-los. Durante a execução do programa, é possível pressionar a tecla *t* para ativar ou desativar os eixos. Também foi implementada a funcionalidade *cameramode* que permite escolher o modo da câmera a ser utilizado. Por padrão, o valor é 1. Por fim, foi adicionada a funcionalidade *polymode*, que altera o modo de desenho entre *GL_FILL*, *GL_LINE* e *GL_POINT*. O modo padrão é *GL_FILL*.

No nodo *translate*, quando este têm a componente de tempo, é possível definir se a linha da trajetória que desenha é visível ou não com a componente *showline* que pode ser *true* or *false*.

No nodo *light* foi implementado o componente de tempo que liga e desliga a luz de x em x tempo.

4 Testes

4.1 Teste 1

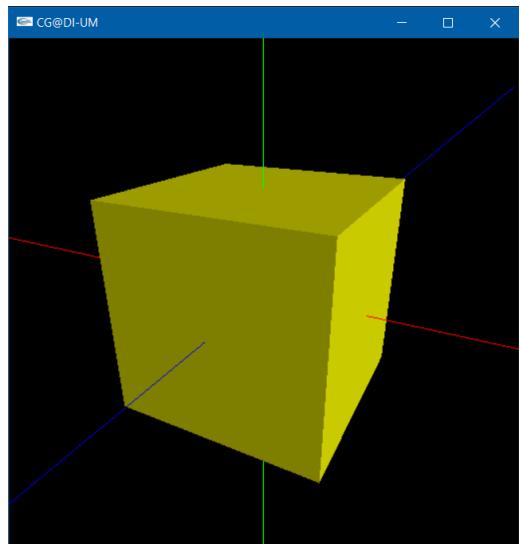
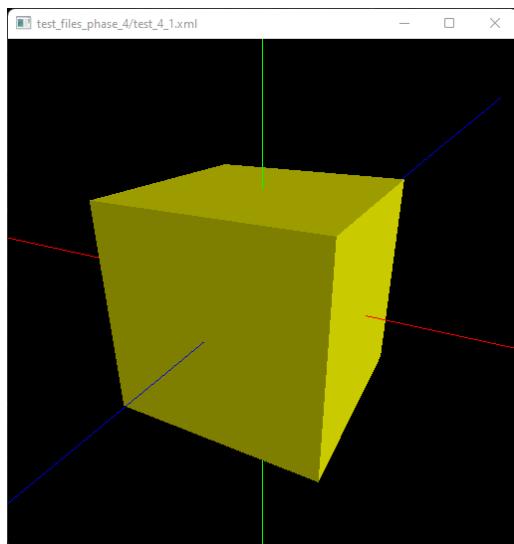
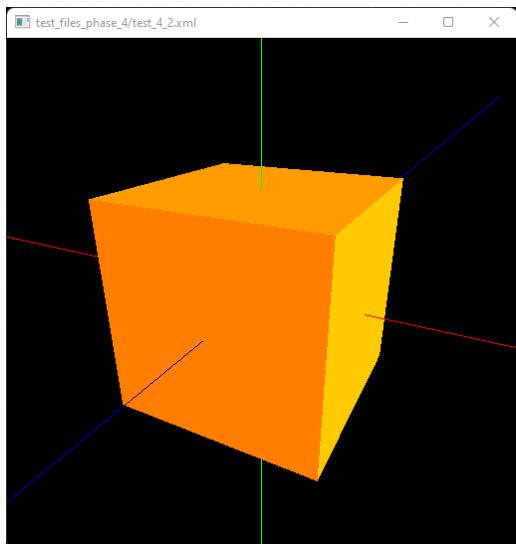
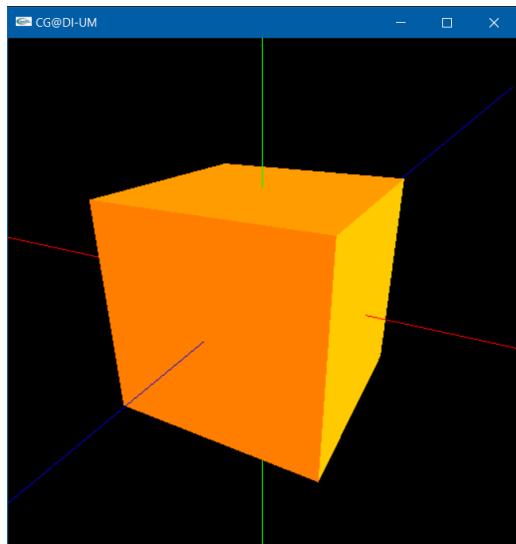


Figura 4.1: Teste 1

4.2 Teste 2



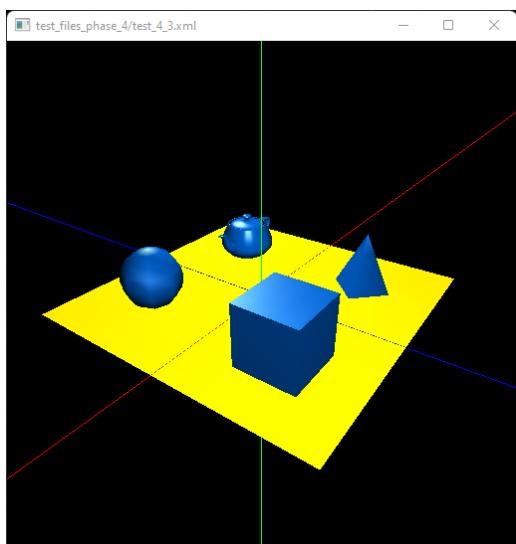
(a) Resultado esperado



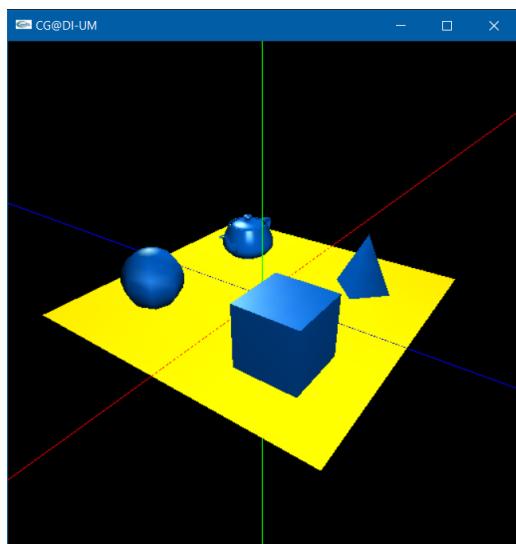
(b) Nosso resultado

Figura 4.2: Teste 2

4.3 Teste 3



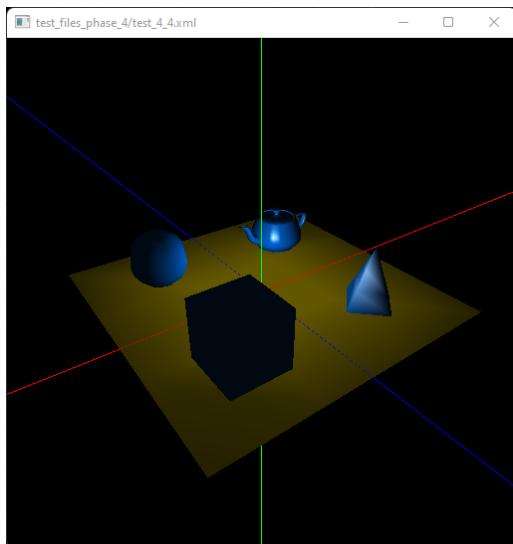
(a) Resultado esperado



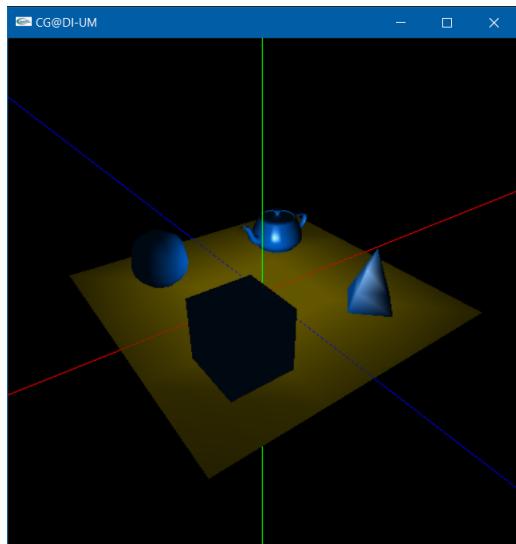
(b) Nosso resultado

Figura 4.3: Teste 3

4.4 Teste 4



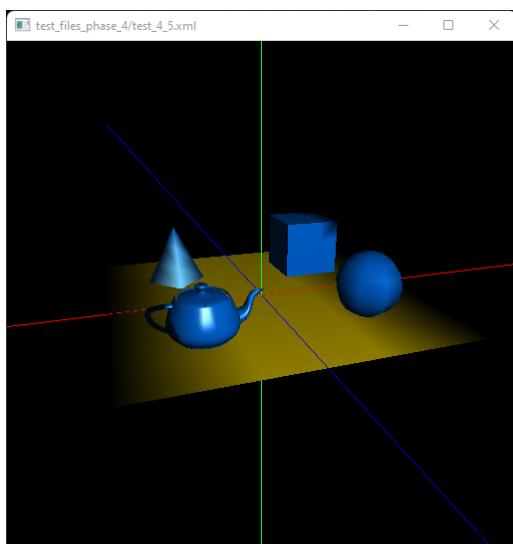
(a) Resultado esperado



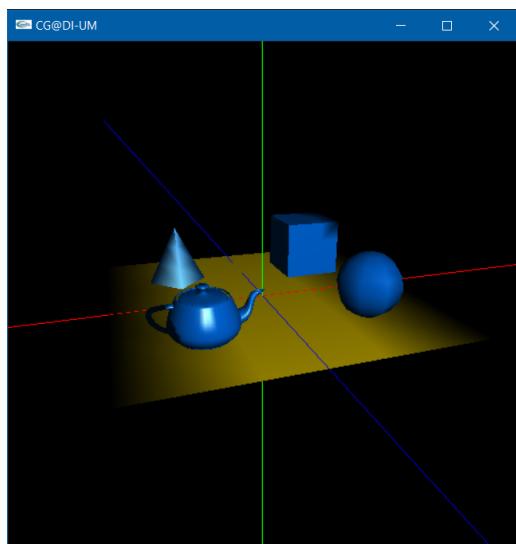
(b) Nosso resultado

Figura 4.4: Teste 4

4.5 Teste 5



(a) Resultado esperado

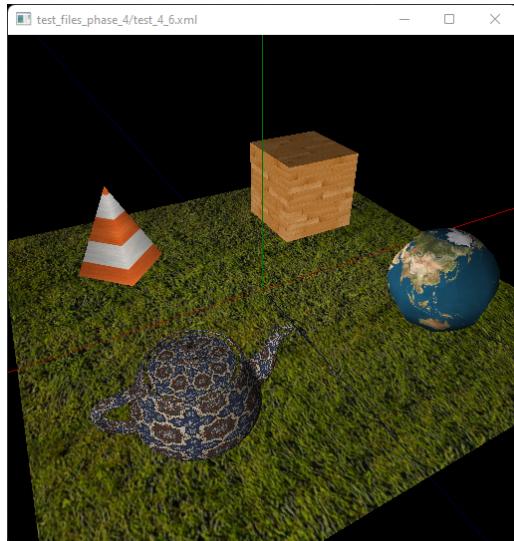


(b) Nosso resultado

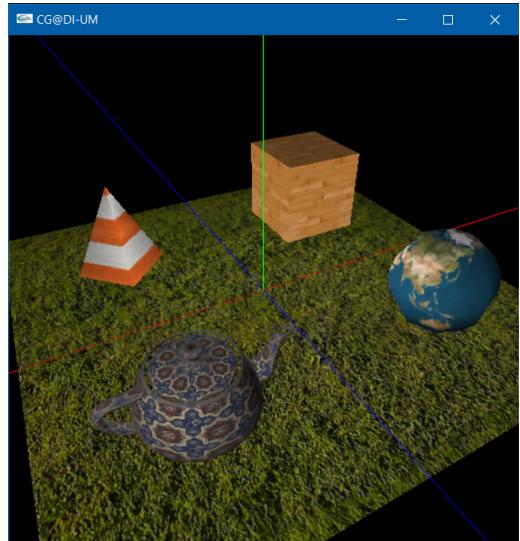
Figura 4.5: Teste 5

4.6 Teste 6

Este é o único teste em que as diferenças são visíveis. Isto deve-se ao facto de ter sido implementado *mipmapping* no projeto, como já referido anteriormente.



(a) Resultado esperado



(b) Nosso resultado

Figura 4.6: Teste 6

5 Demo Scenes

Tal como na fase anterior, o ficheiro de configuração do sistema solar está localizado na pasta engine/config/config.xml, e os modelos estão armazenados na pasta engine/models. A Engine continua a utilizar o ficheiro config.xml da pasta config como ficheiro de configuração e qualquer demo scene a ser utilizado deverá estar localizado na pasta models com esse nome. Os demo scenes também estão todos localizados na pasta 'demoscenes/'.

5.1 Sistema Solar

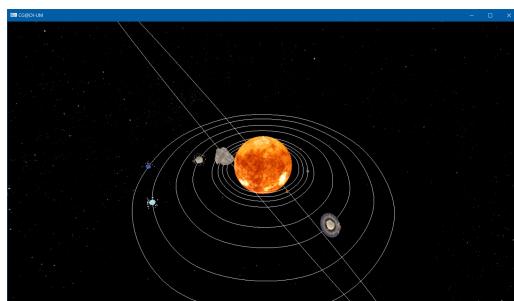
5.1.1 Alterações do sistema solar

Em relação à fase passada foram feitas algumas alterações no Sistema Solar.

Inicialmente, foi inserido um anel em Saturno. De seguida, foi reduzido o raio das luas de alguns planetas de modo a reduzir o tamanho das mesmas. Esta redução foi feita apenas no planetas mais distantes do Sol, mais especificamente Júpiter, Saturno e Urano.

Posteriormente, aplicou-se texturas no cometa, nos planetas e em algumas luas. Apenas foi possível usar texturas na lua da terra, Marte e Neptuno devido ao limite no número das mesmas e pelo facto de não ter sido aplicado a renderização das texturas.

Por fim foi implementado uma skybox com uma texturas de estrelas de modo a tornar visualmente mais atrativo e realista.



(a) Com trajetórias



(b) Sem trajetórias

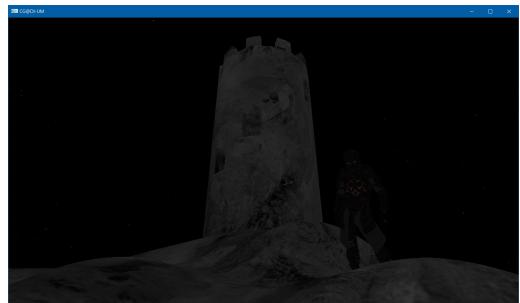
Figura 5.1: Sistema Solar

5.2 Torre com Cavaleiro

Nesta demo scene mostra as capacidades do nosso engine carregar ficheiros .obj e fazer luzes a piscar.



(a) Cena com luz



(b) Cena sem luz

Figura 5.2: Torre

5.3 Cidade

Nesta demo scene foram usados apenas os modelos suportados pelo *Generator* e não se recorreu ao suporte de ficheiros obj implementado nesta fase.



6 Conclusão

Conclui-se que a quarta fase deste projeto foi um sucesso, uma vez que o grupo conseguiu cumprir todos os requisitos propostos no enunciado. Houve melhorias significativas na *Engine* e no *Generator*, permitindo a inclusão de recursos avançados, como iluminação, cores e texturas. Além disso, foram desenvolvidos novos modelos, como o cilindro e a skybox, e o sistema solar foi aprimorado, resultando numa representação mais realista.

Adicionalmente, o grupo implementou extras, como a criação de demoscenes, que agregaram valor ao projeto e permitiram consolidar ainda mais o conhecimento adquirido.