



Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Computação Gráfica

Ano Letivo de 2021/2022

Relatório Fase 4 Normais e Coordenadas de Textura

Grupo

Pedro Araújo	a90614
Pedro Fernandes	a84313
João Cardoso	a94595
Rita Gomes	a87960

Índice

1	Introdução	1
2	Generator	2
2.1	Plano	2
2.1.1	Cálculo das normais	2
2.1.2	Coordenadas de textura	2
2.2	Caixa	2
2.2.1	Cálculo das normais	2
2.2.2	Coordenadas de textura	3
2.3	Esfera	3
2.3.1	Cálculo das normais	3
2.3.2	Coordenadas de textura	3
2.4	Cone	4
2.4.1	Cálculo das normais	4
2.4.2	Coordenadas de textura	4
2.5	Bezier	4
2.5.1	Cálculo das normais	4
2.5.2	Coordenadas de textura	4
3	Engine	5
4	Resultado Final	6
5	Conclusão	10

1 Introdução

O presente relatório serve de suporte ao trabalho realizado no âmbito da unidade curricular de Computação Gráfica. Este trabalho está dividido em quatro fases, sendo que neste relatório será analisada toda a formalização e modelação envolvida na realização da quarta fase.

O objetivo desta fase é modificar a aplicação do generator, fazendo a geração das normais dos vértices para cada primitiva e também das suas coordenadas de textura. É ainda necessário adicionar novos parâmetros aos ficheiros xml, tais como as luzes e o seu tipo e as texturas a adicionar aos modelos.

Para além disso, na aplicação engine foi necessário recorrer a certas alterações no parser de ficheiros xml de modo a preencher novas estruturas de dados com as informações referentes às normais e coordenadas de textura das primitivas lidas.

De seguida, irá ser explicado de que forma foram calculados os pontos das normais e das coordenadas de textura no generator para cada primitiva, bem como a leitura e alterações necessárias no engine.

2 Generator

Nesta fase, o *generator* foi alterado com o objetivo de não só fornecer as coordenadas de cada ponto das primitivas como também gerar as coordenadas normais e as coordenadas de textura de cada ponto. Todos os dados continuam guardados no ficheiro .3d correspondente a cada primitiva (Plano, Caixa, Esfera, Cone, Bezier).

2.1 Plano

2.1.1 Cálculo das normais

Uma vez que o objeto encontrase no plano XZ, os vetores normais serão $(0,1,0)$ para todos os pontos da face voltada para cima e $(0,-1,0)$ para todos os pontos de face voltada para baixo.

2.1.2 Coordenadas de textura

Uma vez que defini-mos que o utilizador poderia indicar o número de divisões por aresta pretendidas, as coordenadas de textura de um dado ponto serão dadas pela partição na qual o vértice se encontra. As coordenadas de textura do plano serão $(i/\text{partition}, 0, j/\text{partition})$, onde i e j representam o número da divisão pela coordenada x e pela coordenada z respetivamente.

2.2 Caixa

2.2.1 Cálculo das normais

Visto que a caixa possui seis faces planas e que para todos os pontos de uma face o vetor normal é o mesmo, concluímos o seguinte:

- O vetor normal dos pontos da face da frente é $(0,0,1)$.

- O vetor normal dos pontos da face de trás é $(0,0,-1)$.
- O vetor normal dos pontos da face da direita é $(1,0,0)$.
- O vetor normal dos pontos da face da esquerda é $(-1,0,0)$.
- O vetor normal dos pontos da face de cima é $(0,1,0)$.
- O vetor normal dos pontos da face de baixo é $(0,-1,0)$.

2.2.2 Coordenadas de textura

Para a caixa, foi necessário definir o template da sua textura. A textura horizontal será $1/4$ e a textura vertical será $1/3$. Isto é, na horizontal, uma face corresponde a $1/4$ da imagem na horizontal, e na vertical, uma face corresponde a $1/3$ da imagem na vertical. Por sua vez, também é necessário considerar o número de divisões por aresta, sendo por isso cada coordenada de textura definida à custa da divisão na qual se encontra.

2.3 Esfera

2.3.1 Cálculo das normais

No caso da esfera, o vetor normal dum dado ponto é igual ao vetor normalizado do vetor que vai do centro da esfera até ao ponto em questão.

2.3.2 Coordenadas de textura

De maneira a calcular as coordenadas de textura da esfera, foi necessário dividir a imagem 2D, pelo número de *stacks* e *slices* que a primitiva possui, considerando que o valor máximo da sua largura e comprimento é 1.

- Uma divisão da esfera é a "interceção" entre uma *stack* e uma *slice*.
- Para cada divisão da esfera, calculámos 6 coordenadas de textura (3 por triângulo).
- O comprimento de uma divisão é $1/slices$.
- A largura de uma divisão é $1/stacks$.
- A cada iteração de *stacks* e *slices* as coordenadas y e x respetivamente, incrementam de acordo com a largura e o comprimento de uma divisão.

2.4 Cone

Para o cone, tivemos em conta o template fornecido para os cilindros, no entanto, desta vez removendo a circunferência superior visto que o cone tem apenas uma na base.

2.4.1 Cálculo das normais

Para a base do cone, que se encontra no plano XZ, o vetor normal é o mesmo para todos os pontos, logo é igual a $(0,-1,0)$. Por outro lado, o vetor normal para qualquer outro ponto da superfície lateral do cone, pode ser obtido da seguinte forma: $(\sin(\alpha), \cos(\text{atan}(h/r)), \cos(\alpha))$.

2.4.2 Coordenadas de textura

No caso do cone, foi apenas utilizado uma imagem correspondente às faces laterais. Para cada iteração pelas slices, podemos determinar as coordenadas de textura, que são: $(j+k, i+k, 0)$, onde i representa o número de stacks, j o número de slices e k pertence $\{0,1\}$, isto é, determina se o ponto encontra-se na stack acima ou na atual.

2.5 Bezier

Com base no nível de tesselação e os atuais índices do ponto, é possível determinar as coordenadas de textura de cada vértice.

2.5.1 Cálculo das normais

No caso dos *patches* de Bezier as normais podem ser calculadas através do produto externo entre as derivadas de u e v , e normalizando o resultado obtido.

2.5.2 Coordenadas de textura

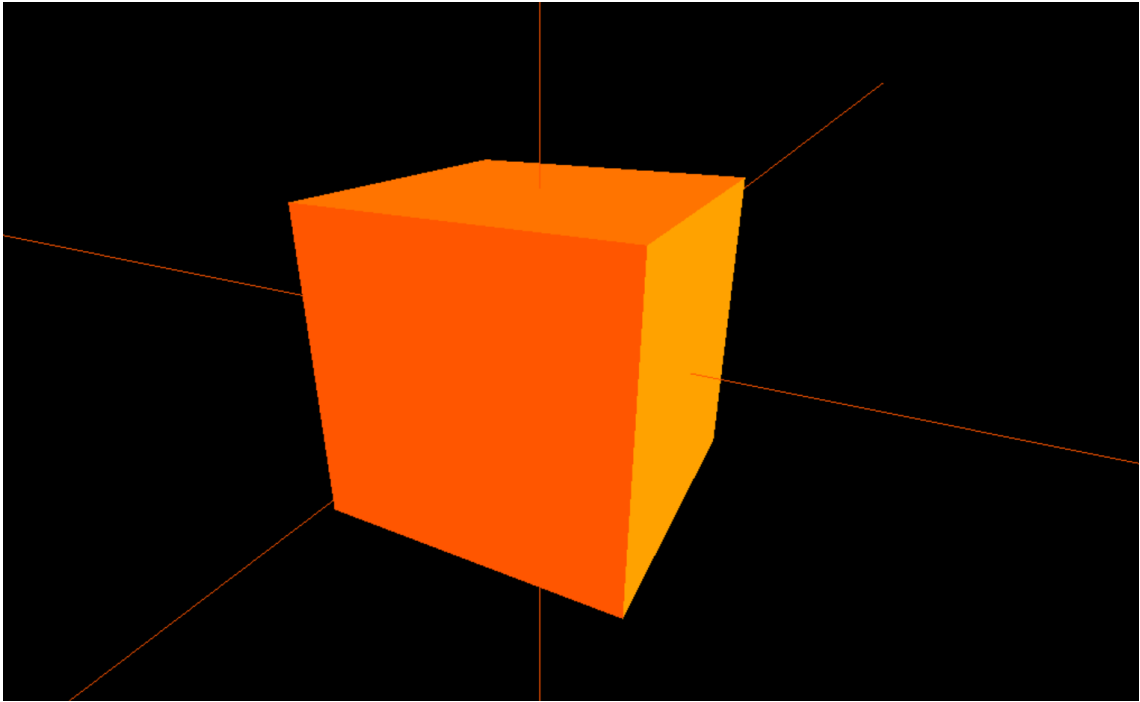
As coordenadas dos pontos de textura são dadas por (v,u) , onde v e u são valores de tesselação, onde é avançado os níveis de tesselação de acordo os índices.

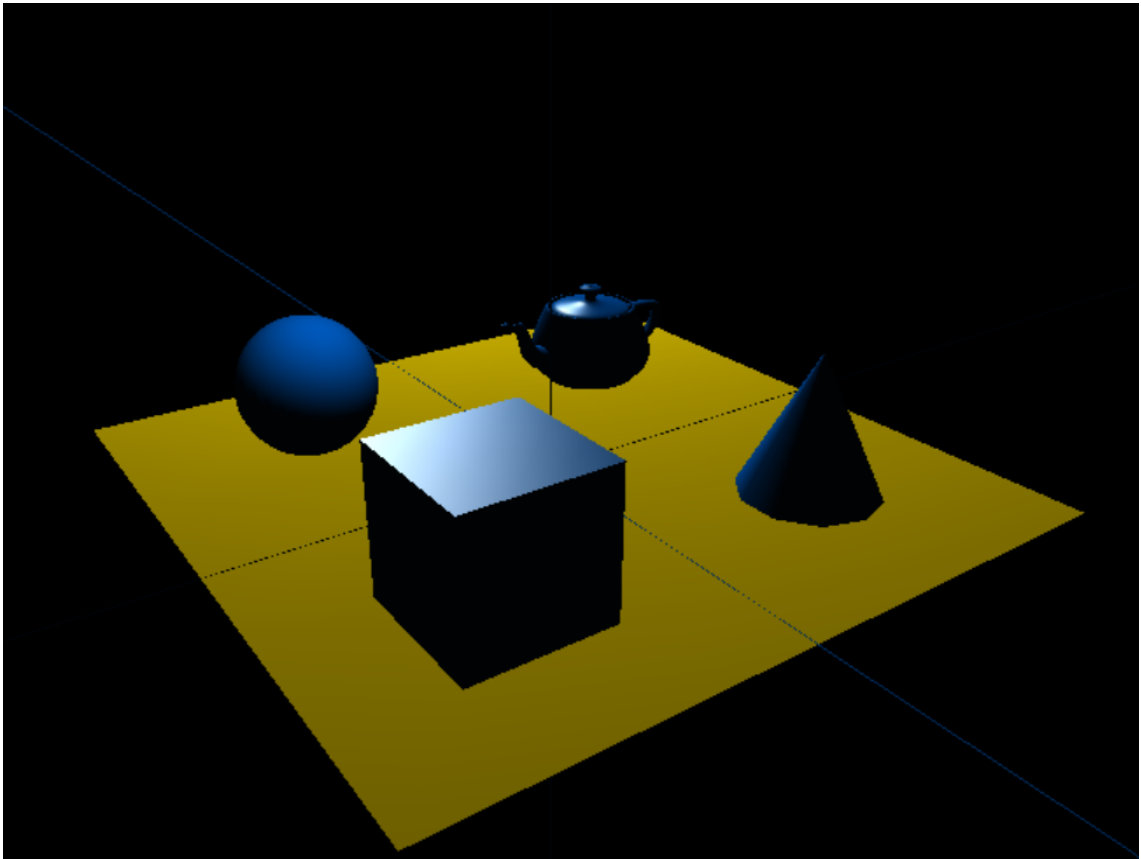
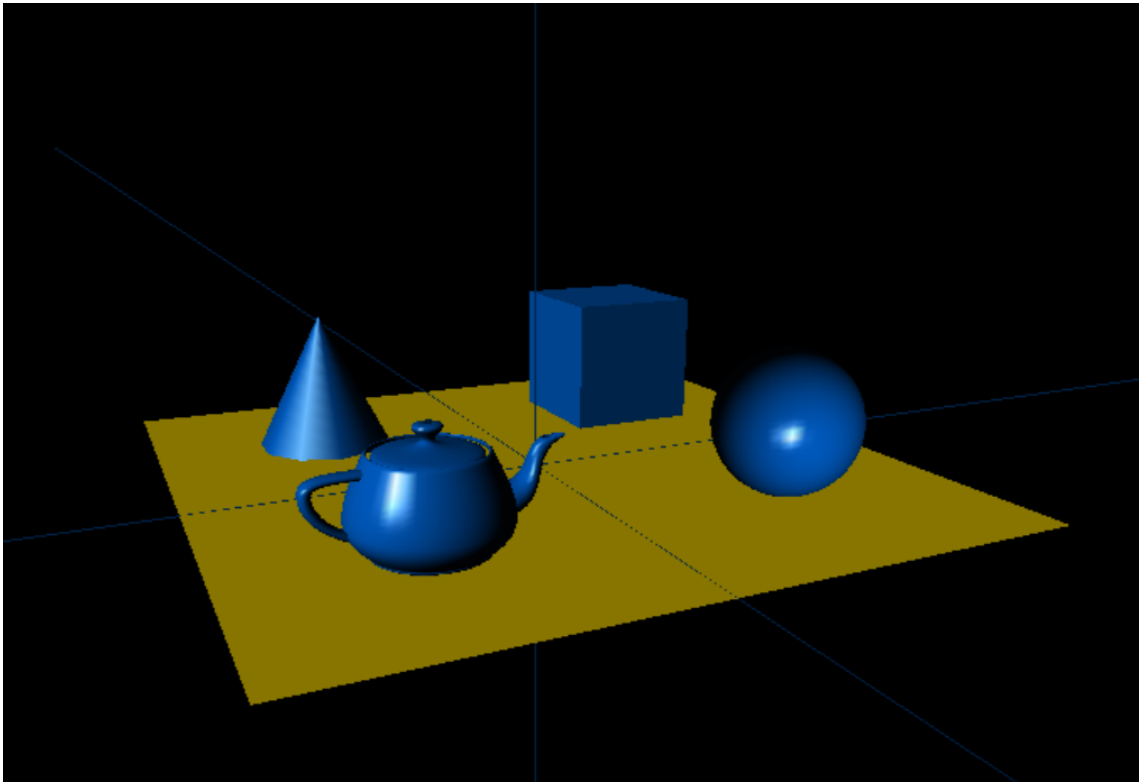
3 Engine

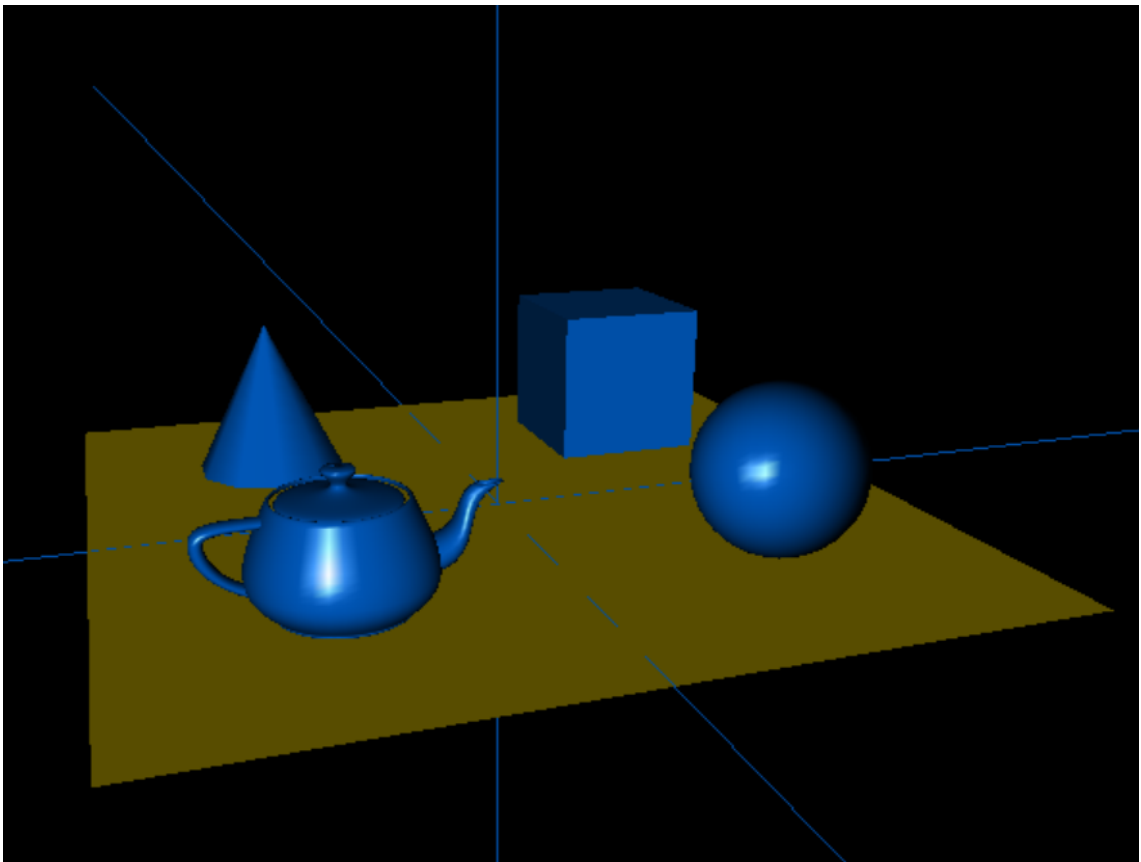
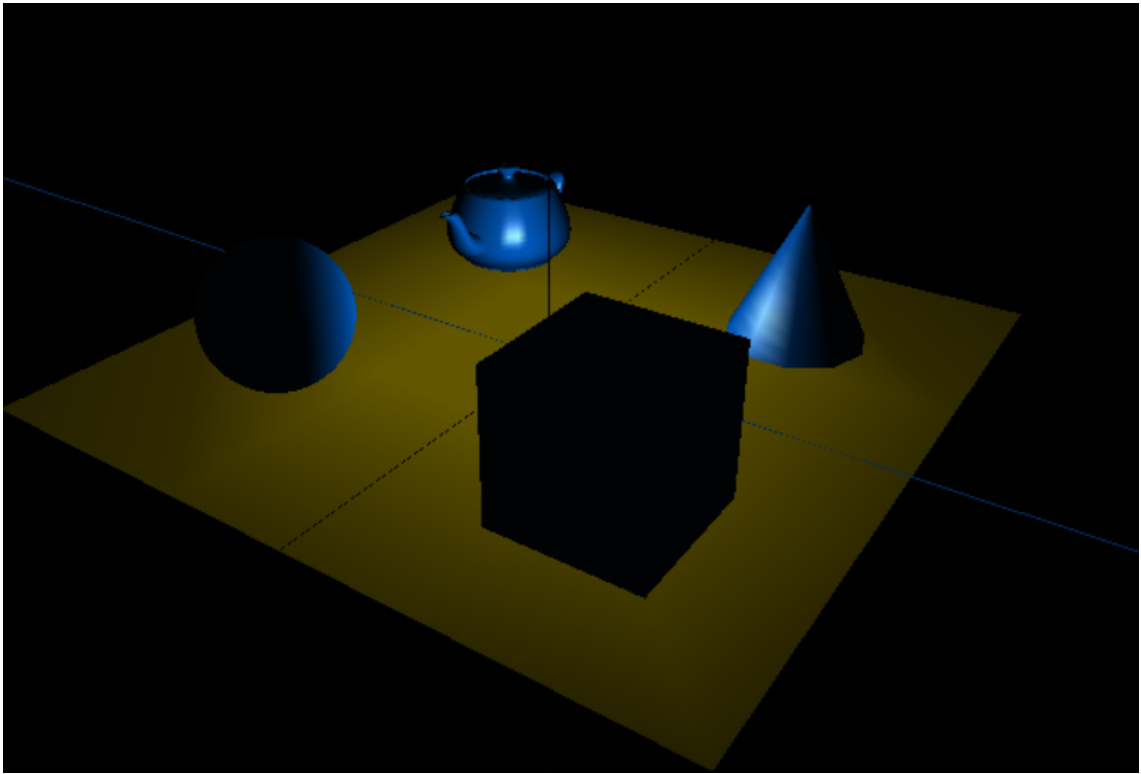
Para a resolução desta fase foi necessário efetuar algumas modificações na maneira como a informação é registada em memória, tal como no modo em que o parsing é realizado, tendo em conta que agora serão passados mais parâmetros dentro do ficheiro *XML*, como é o caso das características de cor e textura.

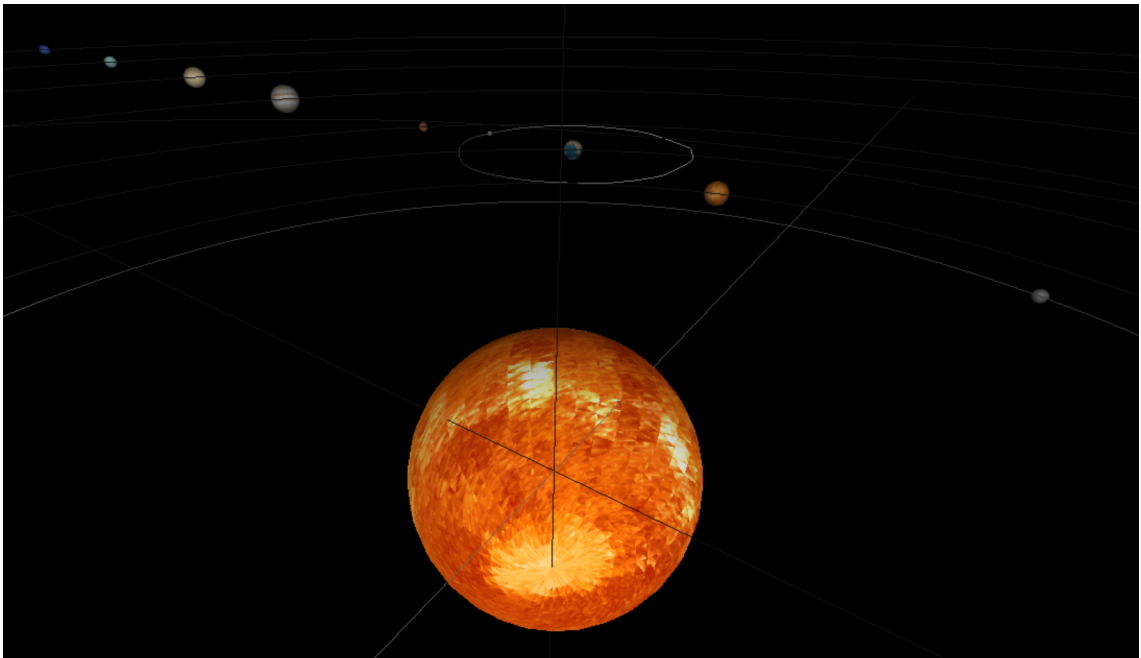
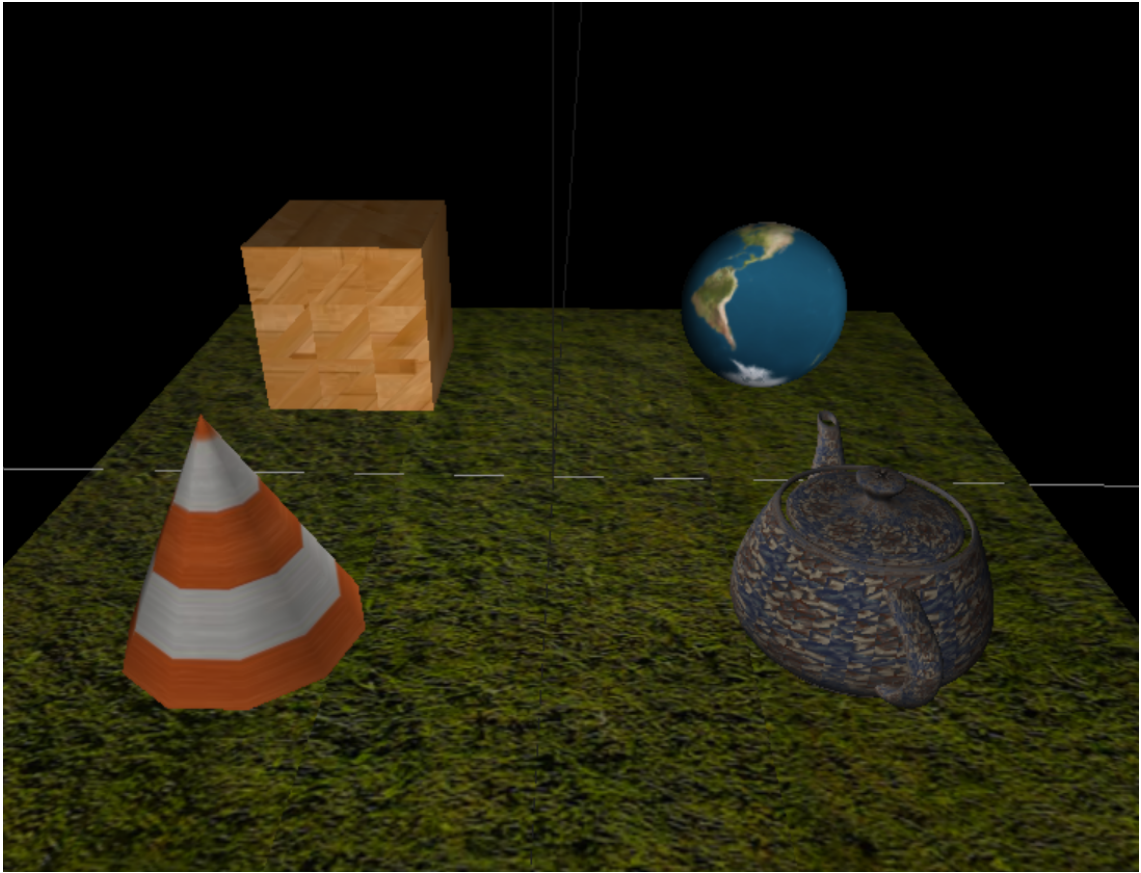
Para além desta mudanças foi necessário preparar o engine para a componente de iluminação e texturas aplicadas aos objetos. Assim foi necessário criar mais 2 vbos para as coordenadas da textura e das normais. Também foi preciso criar novas estruturas para guardar a nova informação, como a struct "Color" que guarda os aspetos relevantes da iluminação e realiza ações sobre elas.

4 Resultado Final









5 Conclusão

Com a elaboração desta fase do projeto, o grupo conseguiu aprofundar conhecimento relativo à iluminação das primitivas e da aplicação das texturas.

Houve desafios que o grupo apresentou maior dificuldade em resolver, como o cálculo das coordenadas das normais e o mapeamento das coordenadas da textura, e também em relação à utilização correta do glut, devil e glew para gerar as figuras pretendidas.

Em suma, efetuando uma revisão geral do trabalho prático, consideramos que o grupo teve uma boa prestação, ultrapassando praticamente todos os desafios propostos.