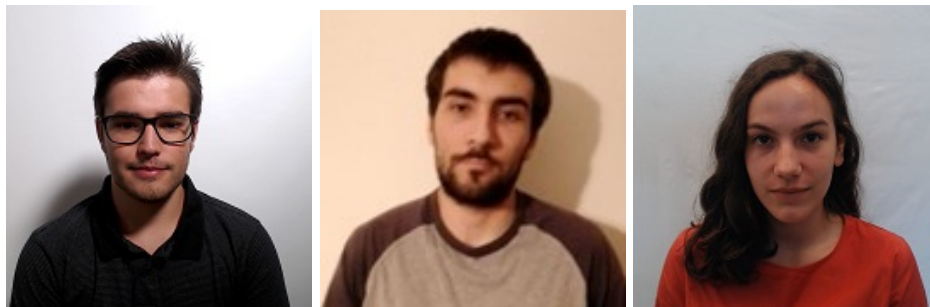


UNIVERSIDADE DO MINHO
LICENCIATURA EM ENGENHARIA INFORMÁTICA

Computação Gráfica - 1^a Fase
Grupo 014

Luís Faria (A93209) Joaquim Roque (A93310)
Gabriela Prata (A93288)

Ano Lectivo 2021/2022



0.1 Introdução

A primeira fase do trabalho prático tinha como objetivo a implementação de duas aplicações. Uma delas é o *generator* que gera os ficheiros com a informação dos modelos, mais precisamente os vértices do modelo, e a outra, é o *engine*, agnóstica à implementação do *generator* que é a aplicação que lê os ficheiros XML de configuração e constrói e mostra os modelos lá referenciados.

0.2 Decisões e abordagens escolhidas

0.2.1 Estrutura

O projeto foi dividido em dois módulos principais, referentes a cada uma das aplicações, cada um contendo uma pasta para o código fonte, outra para os *headers* e o respetivo *makefile*. Decidiu-se ainda que existiria uma outra diretoria contendo código de *libraries* externas e/ou comum a ambas aplicações anteriormente referidas.

Neste último, destacam-se os módulos referentes à representação dos pontos, necessária para ambos os programas e a definição de códigos de erro comuns, de modo a tornar as aplicações mais robustas no tratamento dos mesmos.

Optou-se ainda por usar *makefile* como *build tool*, que irá gerar as aplicações na pasta *bin*.

0.2.2 Generator

Os argumentos passados à aplicação *generator* são tratados adequadamente, garantindo que não existem erros lógicos nos mesmos (e.g. número de slices inferior a 3 ou comprimentos negativos).

O ficheiro criado pela aplicação contém a informação dos modelos em forma de pontos num espaço 3D, que são os vértices do modelo, com as coordenadas dos pontos separadas por ponto e vírgula (;) e um ponto em cada linha.

Arbitrariamente, os pontos são escritos com 30 casas decimais de precisão. De referir ainda que existem dois tipos de pontos, polares e cartesianos. Fazendo uso das características da linguagem C++ (*operator overloading*), podemos escrever um ponto polar para um ficheiro e este será corretamente convertido para cartesiano.

Pegando nos pontos de três em três obtêm-se os triângulos que vão formar o modelo, retirando, assim, necessidade de guardar meta-informação sobre os mesmos.

0.2.3 Engine

Na aplicação *engine*, o parsing do ficheiro XML é feito utilizando a *library* TinyXML2 (guardada na pasta *lib*).

Depois do *parsing* do ficheiro XML, obtêm-se os parâmetros da câmara e a referência dos ficheiros criados pelo *generator* que contém a informação dos modelos. Estes também são parsed

e são obtidos os vértices do modelo que vão ser guardados num vetor de pontos 3D. Cada três pontos consecutivos formam um triângulo.

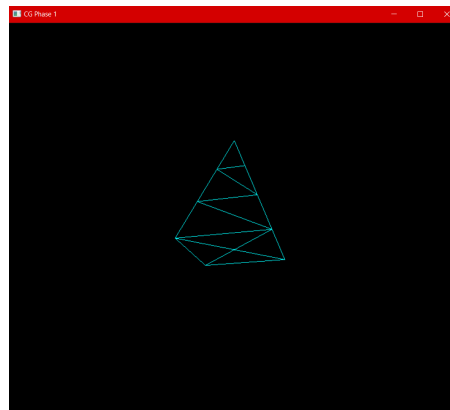
Interação

O grupo decidiu que, para a primeira fase, a interação entre o utilizador e a demonstração do modelo seria pelo teclado. Aqui estão as instruções para tal interação:

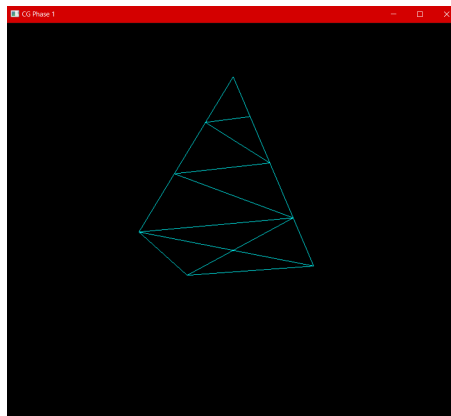
- w - subir a câmara
- s - descer a câmara
- LEFT, RIGHT - rodar a câmara
- f - preencher ou retirar preenchimento dos triângulos
- 1, 2, 3, 4, 5 e 6 - mudar cor
- x - mostrar/esconder eixos

0.3 Testes

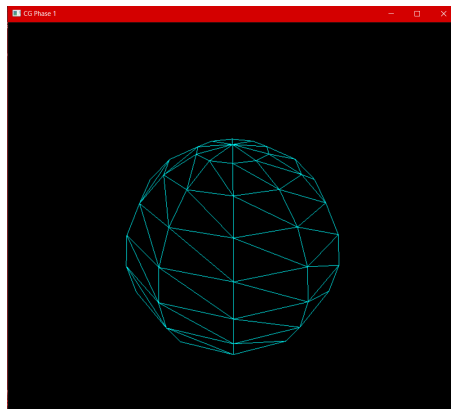
0.3.1 teste_1_1



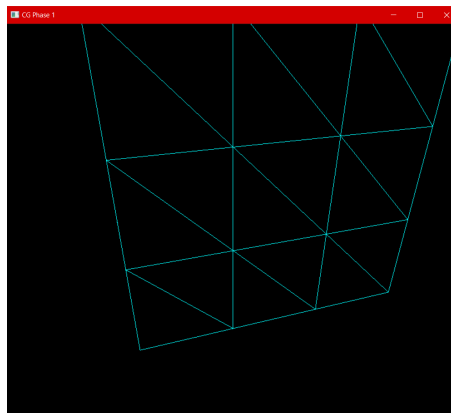
0.3.2 teste_1_2



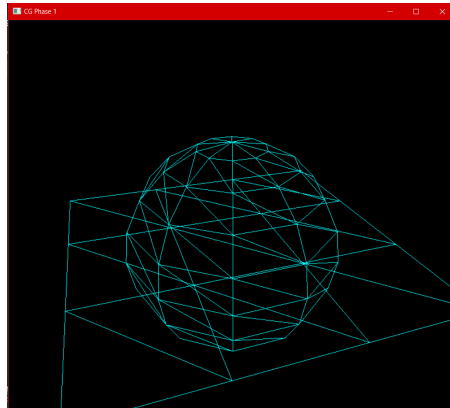
0.3.3 teste_1_3



0.3.4 teste_1_4



0.3.5 teste_1_5



0.4 Conclusão

Com a conclusão desta primeira fase do trabalho prático, podemos afirmar que a implementação desta reforçou a nossa familiarização com os conteúdos de Computação Gráfica já lecionados.

Assim, com uma base que requer pouca ou nenhuma modificação para implementar os requisitos da próxima fase, avançamos para essa, satisfeitos com o alcançado na primeira.