Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia informática
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Computação Gráfica

Ano Letivo de 2023/2024

Computação Gráfica – Checkpoint 2

João Pedro Mota Baptista a100705
João Pedro da Rocha Rodrigues a100896
João Manuel Machado Lopes a100594
Tiago Nuno Magalhães Teixeira a100665

Computação Gráfica

João Pedro Mota Baptista a100705
João Pedro da Rocha Rodrigues a100896
João Manuel Machado Lopes a100594
Tiago Nuno Magalhães Teixeira a100665

8 março 2024

Resumo

No âmbito da disciplina de Computação Gráfica, fomos desafiados a desenvolver uma mini cena sobre uma engine 3D, bem como provar exemplos de utilização para provar o seu potencial.

Em conformidade com essa solicitação, criamos dois módulos principais, um “Generator”, e um “Engine”.

Finalmente, aplicamos, na prática, todas as conceptualizações e ideias previamente definidas na aplicação final.

Nesta segunda fase focamo-nos principalmente na parte do “Engine”, tanto no parsing do ficheiro xml como no desenho das figuras com os novos requisitos.

Área de Aplicação: Desenvolvimento de Software Gráfico
Palavras-Chave: Computação Gráfica

Índice

Generator

A parte do Generator desta segunda fase está bastante semelhante à primeira. Houve apenas uma alteração. Anteriormente, eram utilizados arrays para armazenar as coordenadas dos pontos antes de estas serem escritas nos ficheiros ".3d". Nesta fase foi criada uma struct "Point {float x, y, z}" para armazenar as coordenadas dos pontos em questão, para melhor coerência e organização.
Estes pontos são guardados num " std::vector " para posteriormente serem escritos nos ficheiros pelo "outFile".

Engine

A parte do Engine sofreu mudanças mais significativas.

Primeiramente, a parte relativa ao parsing do xml foi expandida para aceitar novos elementos, como o "translate", "rotate" e "scale". Para além destes, também processa as tags de abertura e fecho dos grupos, de forma a respeitar a hierarquia dos mesmos.
Esta hierarquia de grupos e subgrupos será mantida através de uma stack. Através do uso de uma stack poderemos saber e manipular o grupo que se encontra aberto, pois este corresponderá ao topo da stack. Assim, qualquer transformação encontrada no ficheiro durante o parsing vai ser atribuída ao grupo que se encontra aberto.

Posteriormente foi criado a função "drawGroup". Esta função recebe a stack finalizada, aplica as transformações necessárias e desenhas os modelos através da "drawModel". Através da forma pela qual a "drawGroup" processa a stack final, é possível garantir que as transformações de um grupo são aplicadas às figuras do mesmo e passadas aos respetivos grupos sucessores ("subgroups").

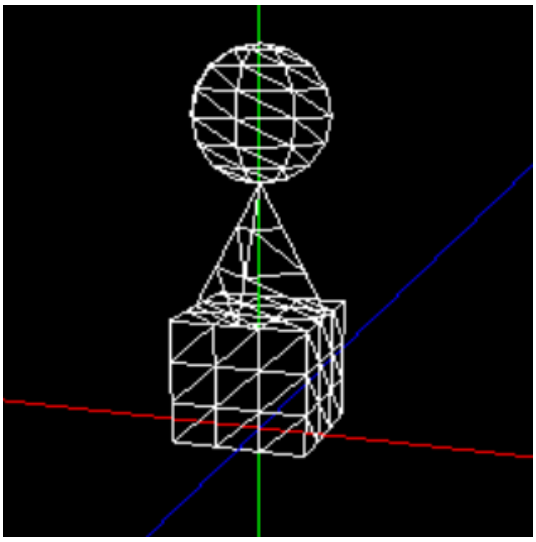
Nesta fase foi também adicionada a opção de movimento através do rato.

Testes

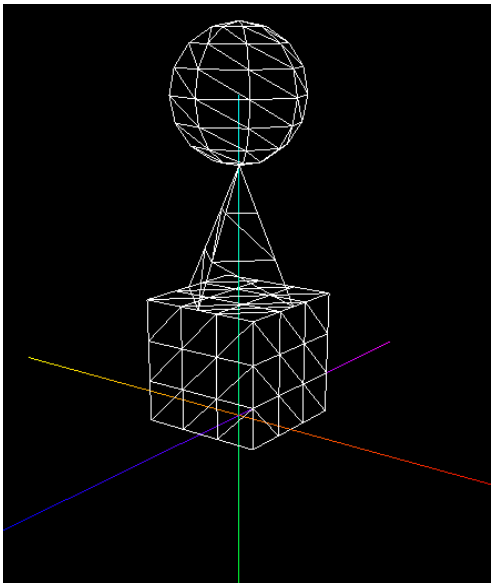
Nesta secção dispomos alguns exemplos de comparações daquilo que era pretendido em relação aos resultados da nossa implementação.

Teste_file 2

Resultado pretendido:

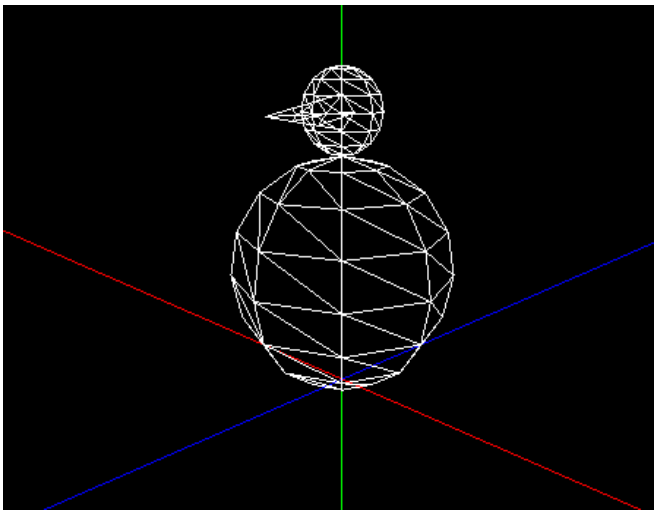


Resultado obtido:

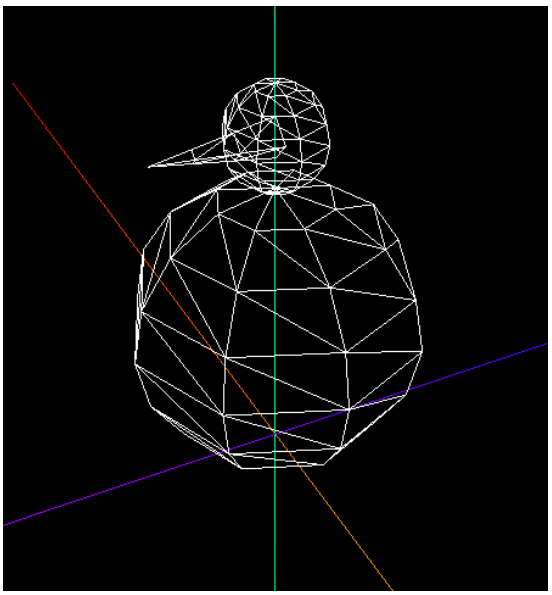


Teste_file 3

Resultado pretendido:

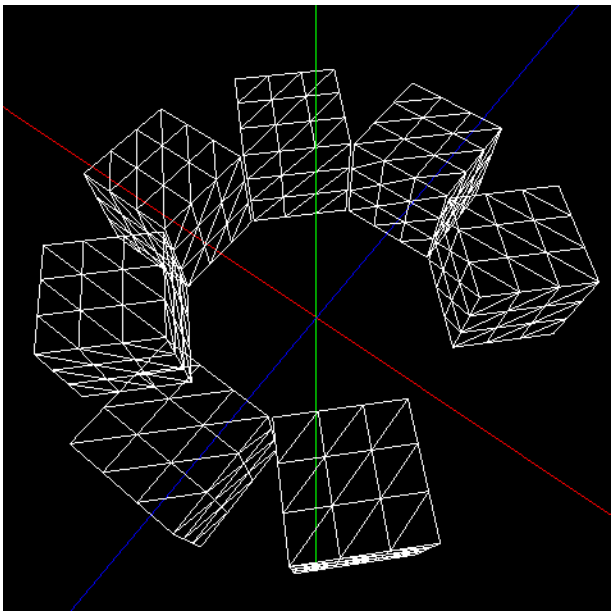


Resultado obtido:

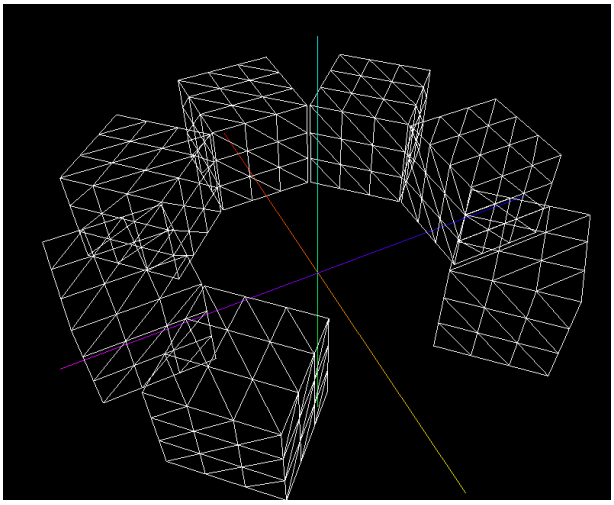


Teste_file 4

Resultado pretendido:



Resultado obtido:



Conclusão

Nesta segunda fase do projeto, pudemos experienciar melhor as transformações a ser aplicadas a um modelo a ser desenhado.

Acreditamos que atingimos o pretendido, a nossa implementação cumpre os testes propostos e cumprimos os requisitos necessários.