Uma imagem com texto, ClipArt

Descrição gerada automaticamente

Universidade do Minho

**Relatório do Projeto Fase 2**

Ano letivo 2021/2022

Abril 2021

Licenciatura em Engenharia Informática

Unidade Curricular de Inteligência Artificial

Uma imagem com pessoa, exterior

Descrição gerada automaticamenteUma imagem com exterior, árvore, pessoa, pose

Descrição gerada automaticamente

Ana Gonçalves a93259 Bruno Pereira a93298

Uma imagem com pessoa, parede, interior, jovem

Descrição gerada automaticamenteUma imagem com pessoa, homem, pose

Descrição gerada automaticamente

Francisco Toldy a93226 João Delgado a93240

Índice

[Introdução 3](#_Toc99806705)

[Torus 4](#_Toc99806706)

[Parser 5](#_Toc99806707)

[Função *Draw* 7](#_Toc99806708)

[Testes 8](#_Toc99806709)

[Teste 1 9](#_Toc99806710)

[Teste 2 9](#_Toc99806711)

[Teste 3 9](#_Toc99806712)

[Teste 4 9](#_Toc99806713)

[Teste 5 9](#_Toc99806714)

[Protótipo de Sistema Solar 9](#_Toc99806715)

[Conclusão 10](#_Toc99806716)

[Anexos 11](#_Toc99806717)

# Introdução

Esta fase do trabalho tinha como objetivo alterar a aplicação do engine de modo a aceitar múltiplas transformações a múltiplos modelos. Para além destas alterações é necessário criar um protótipo de um sistema solar com os modelos criados na primeira fase recorrendo a várias transformações. No entanto foi necessário adicionar um modelo extra ao generator de modo a criar os anéis de alguns dos planetas, torus.

Assim sendo, neste documento vai ser apresentado o modelo extra construído, juntamente com o parser do xml atualizado. Seguidamente vai ser exposto os resultados dos testes fornecidos pelos docentes e o protótipo em companhia do seu ficheiro xml.

Finalmente, encontra-se a conclusão e em anexos alguns pseudocódigos que concordamos ser necessários para uma melhor interpretação do código desenvolvido.

# Primitiva Ring

Considerando todos os modelos existentes da primeira fase, consideramos que foi necessário criar uma que consiga recriar os anéis de planetas, como Júpiter, Saturno, Úrano e Neptuno.

# Parser de XML

Em semelhança ao parser XML da fase anterior, este vai ser chamado na função ***main***de modo a leitura ser feita apenas uma vez, que é constituída por 2 funções principais: ***readCamera***e ***readTransformations****.*

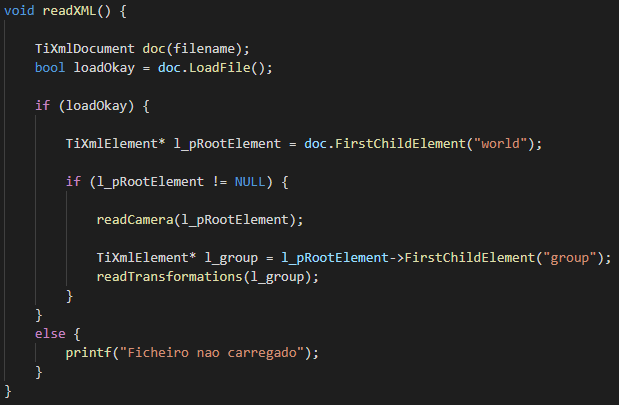
**

Figura 1: Função readXML

***ReadCamera***trata-se do reaproveitamento de uma porção do código da fase anterior que foi separada da função principal para facilitar a leitura de código. Assim sendo, as leituras das características da câmara mantêm-se totalmente igual.

Por outro lado, a função ***readTransformations***, foi criada de raiz com o objetivo de admitir também leitura de transformações (com e sem aninhamento).

Em relação à estratégia utilizada, fizemos uso de um vetor de floats, que durante a leitura do ficheiro XML vai guardar todas as transformações necessárias seguindo o seguinte esquema:

|  |
| --- |
| Insere 1.0f, quando é necessário fazer uma translação e os 3 seguintes valores corresponderá aos valores de x, y e z da translação respetivamente.  Insere 2.0f, quando é necessário fazer uma rotação e os 4 seguintes valores corresponderá aos valores de anglo, x, y e z da rotação respetivamente.  Insere 3.0f, quando é necessário fazer uma escala e os 3 seguintes valores corresponderá aos valores de x, y e z da escala respetivamente.  Insere 4.0f, quando é necessário fazer um pushMatrix, e ocorre sempre que o parser entra num grupo, de modo a conseguir voltar ao estado original antes da transformação.  Insere 5.0f, quando é necessário fazer um popMatrix, e ocorre sempre que o parser sai de um grupo, para voltar ao estado original antes da transformação desse grupo.  Insere 6.0f, quando é necessário fazer a escrita de um modelo. |

Além deste vetor de transformações, criamos um outro, onde vai ser guardado os nomes dos ficheiros dos modelos que têm de ser desenhados.

E assim sendo, estes dois vetores em conjunto permitem o desenho de modelos com as transformações adequadas como é apresentado no seguinte pseudocódigo.

Uma imagem com texto

Descrição gerada automaticamente

Figura 2: pseudocódigo da função readTransformations

# Função *Draw*

Na primeira fase deste trabalho, a função ***draw****,* era utilizada apenas para desenho de múltiplos triângulos, no entanto, foi necessário realizar algumas alterações para possibilitar a existência de transformações geométricas.

Assim sendo, está função basear-se-á nos vetores ***transformations****,* ***models*** e ***vértices****.* Mais especificamente, vai percorrer o vector***transformations***, e em cada elemento identificar a existência de cada uma das transformações ou indicações para ler ficheiro, chamar as funções pushMatrix ou chamar popMatrix.

Esta função traduz-se no seguinte pseudocódigo:

Uma imagem com texto

Descrição gerada automaticamente

Figura 3: Pseudocódigo da função draw

# Testes

## Teste 1

## Teste 2

## Teste 3

## Teste 4

## Teste 5

# Protótipo de Sistema Solar

O conceito proposto pelo enunciado para a demonstração do trabalho feito nesta segunda fase foi a criação do sistema solar. Não estando o programa apetrechado com renderização das várias formas geométricas com cores e texturas, este seria composto por esferas simples representativas dos 8 planetas e o Sol. A esferas base seria então submetida a várias transformações geométricas para criar cada um dos astros na posição correta para formar o sistema solar.

Foi tomada a decisão de desenhar o Sol com centro na origem, posicionando cada um dos planetas ao longo do eixo x.

O modo de raciocínio para as distâncias usadas foi dividido em duas partes

* Escala das esferas

Foi tomada a decisão de fazer um esforço por replicar a escala entre cada um dos astros, exceto o sol. A razão para excluir o sol destes cálculos foi a diferença excessiva entre o seu tamanho comparado a outros planetas (10 vezes maior que Júpiter, o maior planeta), o que tornaria os planetas mais pequenos (Mercúrio, nomeadamente) praticamente invisível na vista de todos os planetas em simultâneo. Assim, os restantes planetas foram escalados com base na seguinte informação retirada do site da NASA:

Uma imagem com texto

Descrição gerada automaticamente

A esfera “base” tem como raio 3.9 unidades, sendo cada uma das escalas aplicadas aos planetas correspondentes à informação mostrada.

A escala escolhida para o sol foi x16 e a escala escolhida para a Lua foi de 0.25, correspondente à escala real Terra/Lua

* Escala das distâncias entre os astros:

Uma vez que a distância relativa entre alguns dos planetas é várias ordens de razão maior que o raio dos mesmos, foi tomada a decisão de ignorar essa escala, optando por fazer um esforço por respeitar apenas a distância relativa entre os atros.

Assim, foi utilizada a tabela seguinte :

Uma imagem com mesa

Descrição gerada automaticamente

Refletindo a tabela no código XML, foi definido que a distância entre a Terra e o Sol seria de 10 unidades no eixo x. Após o desenho de cada um dos planetas seria feita uma translação para igualar a relação tabelada, multiplicada por 10. Assim, por exemplo, entre a Terra e Marte é feita uma translação de 6 (16-10), e entre Úrano e Neptuno é feita uma translação de 192 (388-196).

Após estes cálculos preliminares, foram feitas algumas adaptações de forma ao modelo ficar mais visualmente apelativo.

Por último, foi necessário configurar os valores respetivos à camara no ficheiro XML de forma a facilitar o visionamento do sistema solar.

# Conclusão

Terminada a segunda fase deste trabalho, foram atingidos todos os objetivos propostos pelos docentes. Os resultados obtidos estão maioritariamente em concordância com os resultados indicados no ficheiro de testes fornecido aos alunos e o código apresentado é legível e compreensível com o auxílio do pseudocódigo. Além disso, encontramo-nos bastante satisfeitos com o avanço no desenvolvimento do sistema solar e ansiosos para os progressos que se realizarão nas próximas fases.