Departamento de Eng. Electrotécnica e de Computadores

FCTUC

Relatório

Computação Gráfica

Professor: José Carlos Teixeira (teixeira@mat.uc.pt)

Mecanismo em ambiente 3D

Autores:
João Barreiros
uc2014196880@student.uc.pt
João Ferreira
uc2013139657@student.uc.pt

Data:
5-06-2018

ÍNDICE

1. SUMÁRIO

O objetivo do presente trabalho consiste no desenvolvimento de uma aplicação interativa de um **Mecanismo 3D** composto por, pelo menos, 3 rodas de diâmetros diferentes. A aplicação deverá ser intuitiva e simples de forma a facilitar a sua utilização.

Inicialmente foi feita uma esquematização da aplicação com objetivo de percecionar quais as funcionalidades que devemos implementar e a partir daí obter todos os inputs necessários. Dito isto, enumerámos as seguintes funcionalidades:

1. A rotação das rodas depende dos inputs que o utilizador realizar.

2. A rotação de uma roda influencia a rotação das outras rodas.

3. A aplicação tem 2 modos de operação (automático ou manual).

4. O utilizador pode visualizar o mecanismo de qualquer ângulo.

5. O programa fornece ajuda para facilitar a sua utilização.

Tendo em conta o que foi referido nestes 5 pontos, o presente trabalho tem uma parte dedicada à construção de uma interface gráfica que irá interagir com o utilizador, outra parte dedicada à construção e projecção das rodas, e ainda uma parte dedicada a cálculos que suportam as rodas e a interface. Por fim será necessário realizar os testes necessários à usabilidade, coerência e funcionalidade de todas as implementações elaboradas.

O presente relatório está então dividido nas seguintes secções:

- **Introdução**: Apresentação do tema e análise de requisitos funcionais e não-funcionais.
- **Desenvolvimento**: Proposta de solução e discussão sobre os métodos e estruturas de dados utilizados.
- **Apresentação de uma Sessão de Utilização**: Apresentação da forma de entrada e das diferentes vias de utilização da aplicação com exemplos.
- **Conclusão**: Resumo do que foi feito e de como foi feito, dos resultados, das eventuais dificuldades encontradas e das melhorias e/ou perspetivas futuras.

2. INTRODUÇÃO

2.1 Visão

O presente trabalho tem como tema a elaboração de um programa, em ambiente 3D, que tire proveito das várias primitivas do OpenGL. O objetivo principal consiste no desenvolvimento de uma aplicação interativa de um **Mecanismo 3D** composto por, pelo menos, 3 rodas de diâmetros diferentes. O programa foi desenvolvido na IDE *Microsoft Visual Studio 2017* e a linguagem de programação utilizada foi o C com bibliotecas de OpenGL.

2.2 Análise e Especificação de Requisitos

Define-se um requisito como um comportamento que é esperado do sistema. Durante a fase de análise de requisitos, interpretámos o enunciado para registá-los e categorizá-los em dois tipos distintos: Requisitos funcionais e requisitos não funcionais.

Os requisitos funcionais definem que serviços devem ser fornecidos pelo sistema, que operações podem ser realizadas, quais as reacções a certas entradas, e como deve o estado das entidades constituintes do sistema mudar devido à ocorrência de certas actividades. Apresentamo-los organizados em forma de tabela, e a cada requisito atribuímos uma prioridade de 1 a 3 (onde 1 corresponde a prioridade elevada e 3 prioridade baixa):

Requisitos funcionais	**Prioridade**
A rotação de uma roda deverá influenciar a rotação das outras rodas;	1
A aplicação deverá ter dois modos de operação (automático e manual)	1
Na opção manual, o utilizador deve ter a capacidade de definir a rotação de uma das rodas	1
Sendo um ambiente 3D, o utilizador deve ter a capacidade de visualizar o mecanismo do ângulo de visão que lhe parecer mais adequado;	2
A aplicação deverá ser ergonómica, ter a ajuda necessária para uma fácil utilização.	3

Um requisito qualitativo ou não funcional descreve alguma característica qualitativa que a solução do problema deve ter. Estes requisitos estão relacionados com diversos aspectos, da performance do software à sua segurança, capacidade de utilização, manutenibilidade, precisão, exactidão, custo de desenvolvimento, meta temporal para apresentação do produto final, entre outros. Estes são os requisitos não funcionais que considerámos:

- A aplicação deverá ser simples e intuitiva

- O programa deverá ser robusto e eficiente na implementação

3. DESENVOLVIMENTO

3.1 Conceção

A aplicação foi concebida com o objetivo de ser o mais simples e intuitiva de usar possível, além de funcional. Como tal, desde início planeámos uma interface muito minimalista de forma a não sobrecarregar o utilizador com demasiados componentes no ecrã. Teria de haver alguma espécie de input não só com rato mas também com teclado, de forma a controlar de todas as maneiras possíveis o ângulo de visão e propriedades da câmara/mecanismo, pelo que também arranjámos solução para isso.

3.2 Arquitetura da solução

Tendo em conta os requisitos definidos e mencionados anteriormente, concebemos uma solução para este projecto que é no seu cerne constituída em 2 partes: uma parte de visualização de um mecanismo em ambiente 3D, e uma parte de interacção e controlo em ambiente 2D, a chamada GUI. Estas duas partes estão cada uma no seu *viewport*, mas cada uma depende da outra.

Na primeira parte temos então as 3 rodas dentadas que rodam a uma velocidade constante definida pelo slider de controlo, quando em modo automático, ou rodam segundo a posição do slider que define o angulo de rotação, em modo manual.

Na segunda temos o botão de mudança de modo e o slider de controlo.
Como também é requisito o utilizador conseguir ver o mecanismo do ângulo que lhe parecer mais favorável, adicionámos também essa possibilidade, ao controlar tanto a câmara, como a sua orientação como a orientação do mecanismo, com o teclado, de forma muito intuitiva.

3.3 Interface com o utilizador

Juntamente com os comandos enviados com teclas do teclado que são fornecidos ao utilizador quando este inicia o programa, na linha de comandos, a interface programa-user foi pensada de forma a manter o uso da aplicação o mais intuitivo e simples possível. Desta forma, a GUI é composta por 2 principais componentes: um botão de selecção de modo e um slider, que controla os parâmetros principais da rotação do mecanismo.

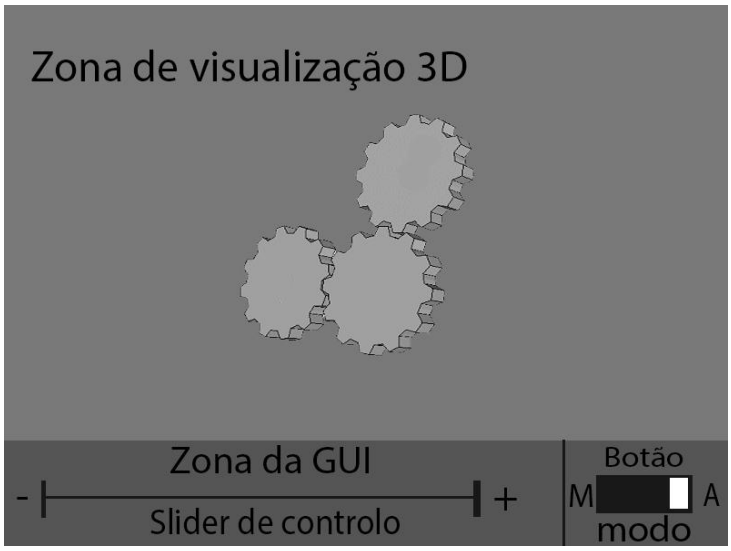


Figura 1- Mockup da GUI

Ao clicar no botão, o quadrado branco no interior do rectângulo preto muda de extremo conforme o modo de funcionamento que nos encontremos. Se estivermos no modo automático este posiciona-se à direita, e à esquerda se no modo manual.

Além desta GUI, o utilizador pode também dar o seu input com certas teclas do teclado, tendo uma vasta série de comandos possíveis que controlam não só a câmara de observação do mecanismo como a própria posição do mesmo. As teclas 'A,S,D,W' controlam a direção para onde a câmara aponta; as teclas 'c' e 'espaço' controlam a posição da câmara relativamente ao mecanismo; as setas do teclado controlam a posição da câmara relativamente aos eixos, e faz esta mover-se de certa forma como um "*pan*". Por fim, as teclas 'I,J,K,L', controladas como se fossem setas, movem o mecanismo, permitindo rodá-lo em todos os eixos.

3.4 Estruturas de dados

Além de estruturas fulcrais ao funcionamento de componentes do programa como teclado e rato, usamos algumas variáveis globais para controlar a posição, orientação e ângulo de visão da câmera, assim como a rotação do mecânismo, os limites dos eixos de coordenas e o tamanho da janela de visualização.

3.5 Principais funções

void MouseButton(int button, int state, int x, int y) **–** Associada a glutMouseFunc(). Esta função é chamada quando clicamos no botão do rato, de forma a permitir executar ações que sejam necessárias aquando o premir do botão. Assim, a função avalia onde foi feito o clique e guarda este ponto para executar certa função.

void Mover(int x, int y) **–** Esta função, associada a glutmotionFunc(), é chamada quando o rato é movido na janela ou quando é pressionado um dos botões do rato.

Além destas duas funções associadas a funções *glut*, utilizamos também as funções *redimensiona_janela, redesenha_cena, tecla_premida, tecla_especial_premida e mostra_cena,* que estão respectivamente associadas a glutReshapeFunc(), glutDisplayFunc(), glutKeyboardFunc(), glutSpecialFunc() e glutIdleFunc(). Estas funções têm todas o comportamento normal esperado das que, por exemplo, utilizamos nas aulas práticas.

Void desenhaUI() **–** Esta função está encarregue de, na sua viewport, criar uma GUI em ambiente 2D que nos permita controlar componentes do ambiente 3D.

Void roda3D(float raio, float largura, int dentes, float tamanho) **–** Esta função vai desenhar uma roda dentada com os atributos passados por parâmetro.

Void desenha() **–** Esta função está encarregue de desenhar o mecanismo 3D e a ui no ecrã.

void tecla_premida(unsigned char tecla, int x, int y) **–** Esta funçao é chamada quando clicamos numa tecla do teclado. Vai permitir contolar a posição e orientação da camera ou então se o utilizador desejar, a rotação do mecanismo em torno de um eixo.

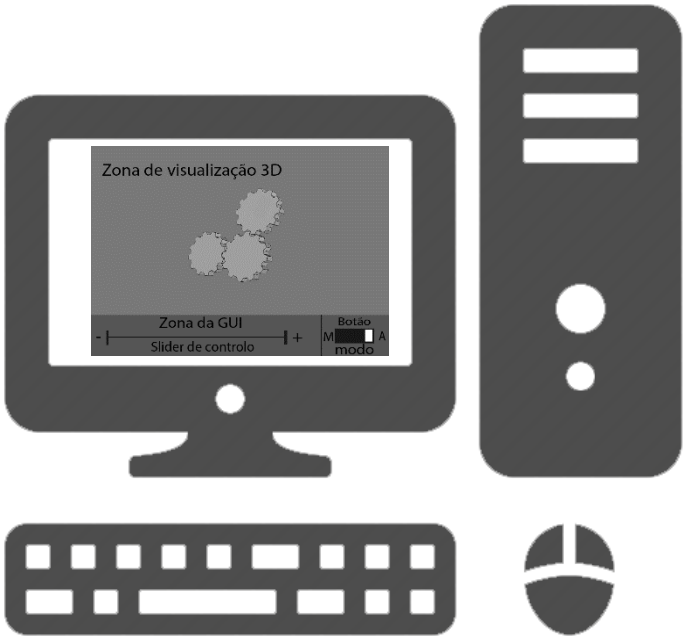
4. APRESENTAÇÃO DE UMA SESSÃO DE UTILIZAÇÃO

4.1 Entrada

Para utilizar o programa, o utilizador pode, como explicado anteriormente, não só utilizar a GUI criada no ambiente gráfico como algumas teclas do teclado que desempenham certas funções. As teclas 'A,S,D,W' controlam a direção para onde a câmara aponta; as teclas 'c' e 'espaço' controlam a posição da câmara relativamente ao mecanismo; as setas do teclado controlam a posição da câmara relativamente aos eixos, e faz esta mover-se de certa forma como um "*pan*". Por fim, as teclas 'I,J,K,L', controladas como se fossem setas, movem o mecanismo, permitindo rodá-lo em todos os eixos.

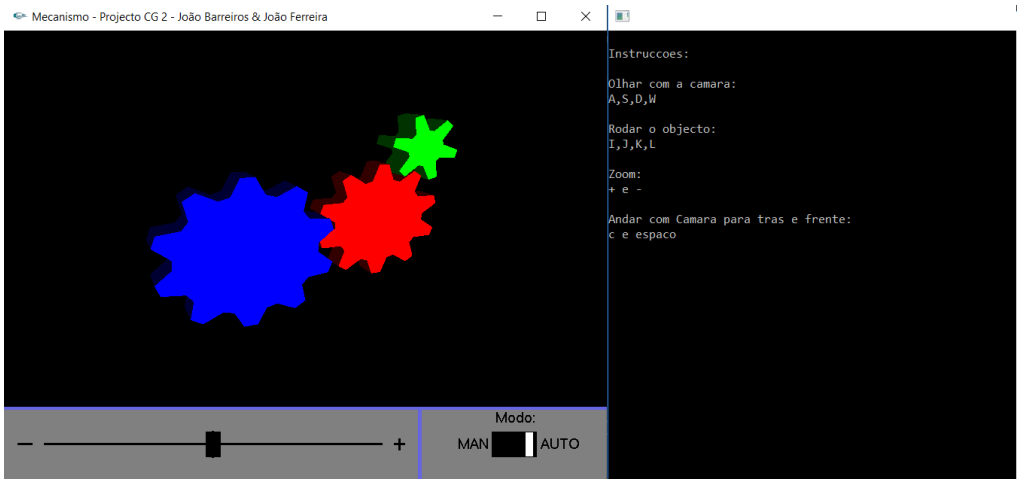
Quanto à GUI, quando o mecanismo funciona em modo automático, o slider de controlo, controlado com o rato, controla assim a velocidade e sentido de rotação (+ positivo ou "para a direita" e – para a esquerda).

Quando este se encontra em modo manual, o próprio slider dá ao mecanismo o ângulo a que ele se encontra nesse momento, ou seja, para o mecanismo rodar, é necessário que o utilizador continue o movimento neste slider.



4.2 Exemplo de Utilização da Aplicação

Quando inicializada a aplicação, são apresentadas ao utilizador as seguintes janelas:



A janela mais à direita é simplesmente uma janela de instruções que o utilizador pode usufruir ao usar a janela principal, a da esquerda.

Na janela principal podemos então observar a zona do ambiente em 3D, com fundo a preto, e a zona da GUI abaixo desta, onde estão os 2 controlos principais: o botão de mudança de modo e o slider de controlo.

Cremos que é um layout bastante intuitivo e que é possível em menos de um minuto o utilizador perceber por completo como funciona o programa.

5. CONCLUSÕES

Acreditamos que a nossa solução ao projecto proposto cumpre todos os requisitos de uma forma simples e funcional. Em contraste ao primeiro projecto, neste tivemos menos dificuldades, maioritariamente por não estarmos a mexer com dados de terceiros, tendo os maiores obstáculos a ver com a parte gráfica/de desenho do programa e não com o armazenamento de dados do mesmo. Tivemos também uma maior facilidade em executar o projecto devido à antecipação que o professor deu nas aulas, quando falou do que ia ser necessário fazer "por alto", o que nos ajudou a preparar e estudar funções que achámos que iriam ser necessárias.

Em suma, cremos que cumprimos tudo o que nos foi pedido e superámos todas as dificuldades que foram aparecendo ao longo da execução do projecto.

6. BIBLIOGRAFIA

Donald Hearn, M. Pauline Baker, W. Carithers, *Computer Graphics with OpenGL,* Prentice Hall, 2013(2004)

Francis S. Hill, Jr., *Computer Graphics using OpenGL,* Prentice Hall, 2006

OpenGL® 2.1, GLX, and GLU Reference Pages - https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/