

# **Universidade da Beira Interior**

## **Departamento de Informática**



**Departamento de  
Informática**

**Licenciatura em Engenharia Informática**  
**Unidade Curricular: Computação Gráfica**

**Projeto Solar System**

Elaborado por:

**Tiago Almeida - a48278**  
**João Rainha - a48506**

Orientador:

**Professor Doutor Abel Gomes**

Covilhã, 8 de janeiro de 2024



# ***Agradecimentos***

Queremos expressar nossos agradecimentos ao Professor Doutor Abel Gomes, o nosso orientador, pela constante justiça e lealdade às suas convicções ao propor o tema deste projeto. A sua orientação desempenhou um papel crucial no desenvolvimento deste trabalho, assim como as abordagens compartilhadas em sala de aula, tanto nas aulas práticas como nas teóricas. A forma eficaz de como esses conhecimentos foram transmitidos teve um impacto significativo no nosso crescimento. Sentimos que evoluímos bastante e isso se deve em grande parte à valiosa contribuição do Professor Doutor Abel Gomes.



## ***Resumo***

O propósito deste projeto é a concepção de um sistema solar virtual, empregando tecnologias com três dimensões. Buscamos forjar uma experiência genuína, envolvente e participativa do nosso sistema solar, permitindo aos usuários a oportunidade de explorar e imergir nas maravilhas do universo. Para além disso, será integrado uma componente educativa que abrange informações sobre os distintos planetas e outras características do sistema solar. Deste modo, temos como objetivo oferecer uma experiência singular e estimulante, não apenas para os professores, mas também para os demais estudantes.



# Conteúdo

<b>Conteúdo</b>	<b>v</b>
<b>1 Introdução</b>	<b>9</b>
1.1 Âmbito e Enquadramento . . . . .	9
1.2 Motivação UBI . . . . .	9
1.3 Objetivos . . . . .	10
1.4 Abordagem . . . . .	10
1.5 Organização do documento . . . . .	11
<b>2 Estado de arte, desenvolvimento e aplicações</b>	<b>13</b>
2.1 Introdução . . . . .	13
2.2 Overleaf . . . . .	13
2.3 Discord . . . . .	14
2.4 GitHub . . . . .	14
2.5 Visual Studio Code . . . . .	14
<b>3 Ferramentas e Tecnologias utilizadas</b>	<b>17</b>
3.1 Introdução . . . . .	17
3.2 OpenGL . . . . .	17
3.3 GLM . . . . .	18
3.4 GLFW . . . . .	18
3.5 Glad . . . . .	18
3.6 Freetype . . . . .	18
<b>4 Etapas de desenvolvimento</b>	<b>19</b>
4.1 Introdução . . . . .	19
4.2 Divisão das tarefas . . . . .	19
4.3 Procedimentos . . . . .	20
<b>5 Implementação</b>	<b>21</b>
5.1 Introdução . . . . .	21
5.2 Dependências . . . . .	21
5.3 Implementações . . . . .	22

5.3.1	Planetas . . . . .	22
5.3.2	Céu/Skybox . . . . .	22
5.3.3	Orbitas dos Planetas . . . . .	23
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>25</b>
6.1	Conclusões . . . . .	25
6.2	Trabalho Futuro . . . . .	25
6.3	Bibliografia . . . . .	26



## ***Lista de Figuras***



## ***Lista de Tabelas***

Tabela 5.1 - imports utilizados



## ***Palavras-Chave***

3D, Computação Gráfica, Projeto, OpenGL, Sistema Solar



# ***Acrónimos***

CG - Computação Gráfica  
CAD - Computer Aided Design  
EI - Engenharia Informática  
IDE - Integrated Development Environment  
OpenGL - Open Graphics Library  
UBI - Universidade da Beira Interior  
UC - Unidade Curricular





## **Capítulo**

# **1**

## **Introdução**

### **1.1 Âmbito e Enquadramento**

O presente relatório foi elaborado por alunos do terceiro ano da Licenciatura de Engenharia Informática (EI) da Universidade da Beira Interior (UBI). O mesmo foi desenvolvido no contexto da Unidade Curricular (UC) de Computação Gráfica (CG) da UBI, através de pesquisas por parte dos alunos e material disponibilizado pelo professor da própria UC.

### **1.2 Motivação UBI**

A UC de Computação Gráfica constitui um campo de estudo que viabiliza a produção de imagens e animações de alta qualidade, oferecendo a capacidade de representar objetos e cenários de maneira minuciosa. O Sistema Solar, por sua vez, é um tema cativante, e a possibilidade de visualizá-lo em três dimensões amplia significativamente a nossa compreensão e conhecimento acerca deste conjunto celestial. Além do fascínio proporcionado, a criação de um modelo de três dimensões do sistema solar também se revela como uma valiosa ferramenta educacional. Tal abordagem permite que as pessoas tenham uma representação mais exata e clara do funcionamento do nosso sistema solar. Desta forma, a realização deste projeto de Computação Gráfica dedicado ao Sistema Solar em 3D não apenas representa uma oportunidade perfeita para aprofundar nosso entendimento sobre este tema, mas também contribui para enriquecer o conhecimento e a compreensão do público em relação ao mesmo.

## 1.3 Objetivos

Este projeto visa desenvolver um sistema solar interativo, abrangendo tanto representações em 2D quanto em 3D. Para concretizar essa proposta, a linguagem escolhida é o C++, com o suporte das bibliotecas OpenGL, GLFW, GLEW. Algumas características essenciais da aplicação gráfica incluem:

- Criar modelos tridimensionais para o Sol, os planetas e seus satélites, incluindo a Lua em relação à Terra;
- Implementar menus interativos que ofereçam informações detalhadas sobre os elementos do sistema solar;
- Aplicar texturas aos planetas para aprimorar o seu realismo;
- Permitir movimento livre da câmera em qualquer posição do sistema solar, afetando a incidência de luz sobre os elementos;
- Incorporar zoom e ajuste de perspectiva utilizando o teclado;
- Calcular e representar as sombras geradas pelos planetas e satélites em relação uns aos outros.

## 1.4 Abordagem

De maneira a abordar este tema em Computação Gráfica, decidimos propor os seguintes passos:

1. Pesquisar informações sobre órbitas e movimentos, constituição e características sobre os planetas, etc;
2. Realizar a criação dos planetas e dos corpos celestes, usando as informações recolhidas na pesquisa realizada anteriormente;
3. Aplicar as texturas dos vários corpos celestes, de maneira a torná-los realistas;
4. Criar animações das órbitas e dos movimentos dos planetas em redor do Sol;
5. Realizar testes e experiências de maneira a confirmar o seu funcionamento e posteriormente realizar um relatório, apresentando uma descrição pormenorizada sobre a criação deste projeto.

## 1.5 Organização do documento

1. **Introdução** - Introduz o tema do projeto realizado e que nos foi proposto, bem como os objetivos e a organização do documento em questão;
2. **Estado de arte, desenvolvimento e aplicações** - descreve as aplicações e ferramentas utilizadas para fazer e construir este projeto;
3. **Ferramentas e Tecnologias utilizadas** - É descrita as bibliotecas e funcionalidades externas que foram utilizadas para a realização deste projeto;
4. **Etapas de desenvolvimento** - Descreve as etapas minuciosamente que foram feitas para construir o programa e também inclui detalhes sobre a implementação de código;
5. **Testes e Validação** - Apresenta os testes realizados durante a implementação do programa;
6. **Conclusões e Trabalho Futuro** - Faz uma breve conclusão sobre este tema e sobre o projeto que foi trabalhado.



## Capítulo

# 2

## ***Estado de arte, desenvolvimento e aplicações***

### **2.1 Introdução**

Este capítulo descreve as aplicações e ferramentas que foram usadas para realizar e desenvolver este projeto, fornecendo uma descrição detalhada sobre cada uma das ferramentas abordadas nesta parte do relatório:

- secção 2.2 - **Overleaf**, ferramenta utilizada para produzir o relatório;
- secção 2.3 - **Discord**, plataforma que serviu como meio de comunicação;
- secção 2.4 - **Visual Studio Code**, aplicação fundamental para o desenvolvimento do programa;
- secção 2.5 - **Github**, plataforma de partilha de código.

### **2.2 Overleaf**

O overleaf é uma ferramenta online que usamos como base para elaborar este relatório e para a escrita deste documento. Disponibiliza recursos e opções de maneira a facilitar a criação deste relatório e existe uma enorme diversidade de opções que permitem melhorar o texto. Oferece imensas opções para o utilizador, sendo que uma das principais é a possibilidade de trabalhar em conjunto ao mesmo tempo. é bastante útil devido ao facto de possibilitar a

partilha de ideias e uma maior reentabiização don tempo pois podem estar a trabalhar mais que uma pessoa no documento. Podemos atualizar constantemente o projeto em que trabalhamos, desta forma podemos detetar erros e melhores opções para melhorar a parte estética do documento produzido.

## **2.3 Discord**

O Discord é uma plataforma online amplamente adotada por grupos de amigos, equipas e diversas comunidades. A sua utilidade reside na capacidade dos utilizadores poderem criar canais para comunicação via texto e voz, proporcionando interações eficientes. Além disso, a plataforma oferece ferramentas de moderação e configurações de privacidade, possibilitando aos administradores manterem o controle sobre o conteúdo e comportamento. A flexibilidade do Discord se destaca com uma variedade de integrações, sendo que se conecta a diferentes plataformas e serviços, como jogos, música e streaming. Esta diversidade torna o Discord num recurso valioso para comunidades em geral. A funcionalidade de partilha de ecrã adiciona outro nível de interatividade, contribuindo para uma comunicação mais rica. O aplicativo é uma escolha popular para comunicação tanto por voz quanto por texto, oferecendo a vantagem adicional de ser gratuito, sem limitações de tempo para as chamadas de voz. Essa acessibilidade promove interações sem interrupções entre os utilizadores.

## **2.4 GitHub**

O GitHub desempenha um papel fundamental como um serviço online dedicado à gestão do código-fonte e colaboração. A sua função principal é proporcionar um ambiente para armazenar e administrar projetos, tanto de código público quanto privado. Os utilizadores têm a capacidade de criar repositórios de código, realizar o controlo de versões, adicionar colaboradores aos projetos e monitorar os problemas e tarefas associadas. Além disso, o GitHub oferece uma gama de ferramentas que incluem a revisão de código e integração contínua. Esta abordagem integrada facilita o processo de desenvolvimento, promovendo eficiência e qualidade nos projetos.

## **2.5 Visual Studio Code**

Trata-se de um m Integrated Development Environment (IDE) que proporciona uma extensa gama de recursos e ferramentas, agilizando o processo de

criação, depuração e publicação de aplicativos para os programadores. Em resumo, algumas das características suas características principais incluem um suporte abrangente para várias linguagens de programação, funcionalidades robustas de depuração e teste e ferramentas de design integradas, facilitando a criação de interfaces atrativas e intuitivas para os desenvolvedores.





## **Capítulo**

# 3

## ***Ferramentas e Tecnologias utilizadas***

### **3.1 Introdução**

Neste capítulo serão apresentadas e listadas as diversas ferramentas que desempenharam um papel crucial ao longo do nosso projeto. Estas foram fundamentais para o desenvolvimento do código que viabilizou a criação do Sistema Solar.

### **3.2 OpenGL**

A Open Graphics Library (OpenGL) é uma interface de programação de aplicações (API) que opera em diversas plataformas e suporta várias linguagens. Ela é dedicada à renderização de gráficos vetoriais em 2D e 3D, utilizada para interagir com a unidade de processamento gráfico (GPU) e obter renderização acelerada por hardware. Amplamente usada em áreas como Computer Aided Design (CAD), realidade virtual, visualização científica e jogos virtuais, o OpenGL é especialmente valioso para programadores de jogos, pois aproveita a renderização acelerada por hardware. Esta API oferece aos desenvolvedores um conjunto diversificado de comandos para renderizar uma ampla gama de objetos gráficos, incluindo pontos, linhas, triângulos e quadrados.

### 3.3 GLM

A OpenGL Mathematics (GLM) é uma implementação em C++ destinada a proporcionar eficiência e facilidade no uso na programação de gráficos 3D. Projetada para oferecer uma ampla gama de objetos e funções matemáticas, a GLM permite a criação e manipulação de objetos tridimensionais. As suas funcionalidades abrangem operações em vetores e matrizes, entre outras. Um aspecto que distingue a GLM é sua compatibilidade com a API OpenGL, o que a torna aplicável em qualquer software baseado em OpenGL.

### 3.4 GLFW

A Framework de Biblioteca Gráfica (GLFW) é uma biblioteca gratuita e de código aberto que opera em diversas plataformas. A sua API simplificada possibilita a criação e gestão de janelas, contextos (OpenGL) e superfícies, além de lidar com entradas e eventos. Desenvolvida em linguagem C, a GLFW é compatível com os sistemas operativos Windows, MacOS e Linux. Essa versatilidade torna a GLFW uma escolha viável em diversas aplicações, incluindo jogos, realidade virtual e em outras áreas.

### 3.5 Glad

A Glad representa uma biblioteca essencial para os programadores, possibilitando o uso eficiente do OpenGL em múltiplas plataformas. Ao utilizar a biblioteca Glad, os desenvolvedores têm a capacidade de estabelecer contextos OpenGL, administrar ponteiros de funções relacionadas ao OpenGL, carregar funções específicas e gerir estados associados ao OpenGL. Esses atributos são particularmente valiosos, dada a natureza independente da plataforma do OpenGL, simplificando assim a tarefa de escrever código que funcione de maneira consistente em várias plataformas.

### 3.6 FreeType

O FreeType é uma biblioteca que possibilita que as aplicações realizem a renderização de texto em fontes bitmap e vetoriais. Amplamente usado por diversos sistemas operativos e aplicações, o FreeType desempenha um papel essencial na exibição de fontes na tela. Além disso, é frequentemente utilizado por editores de fontes para diversas finalidades.

## **Capítulo**

# 4

## ***Etapas de desenvolvimento***

### **4.1 Introdução**

Neste capítulo, abordaremos o processo de criação do programa em OpenGL dedicado à visualização em três dimensões do sistema solar. Vamos explorar a divisão das várias tarefas entre os membros da equipa e detalharemos as fases cruciais do desenvolvimento do programa.

### **4.2 Divisão das tarefas**

A divisão de tarefas foi realizada da seguinte forma:

- Criação de Planetas - João Rainha
- Órbitas - Tiago Almeida
- FreeType - João Rainha
- Transição da Câmara - Tiago Almeida
- Texturas - João Rainha
- SkyBox - Tiago Almeida
- Relatório - João Rainha e Tiago Almeida

### 4.3 Procedimentos

Na criação deste projeto foi preciso seguir as seguintes etapas/procedimentos:

1. Estabelecer as dimensões e proporções do sistema solar, abrangendo a definição do tamanho dos planetas em relação ao Sol e as distâncias entre eles.
2. Criar representações visuais dos planetas, optando por formas geométricas simples como esferas;
3. Enriquecer a aparência dos planetas adicionando texturas realistas através do uso de imagens disponíveis no link fornecido no enunciado do projeto.
4. Incorporar a Lua ao planeta Terra, seguindo o mesmo procedimento utilizado para desenhar os próprios planetas.
5. Introduzir movimento ao cenário, utilizando rotações e translações para simular os deslocamentos dos planetas em torno do Sol e o movimento lunar ao redor do planeta Terra.
6. Aprimorar a visualização do cenário com iluminação e sombras, aplicando técnicas como o Phong shading e utilizando fontes de luz adequadas.
7. Permitir uma variedade de posições para a câmera, incluindo a capacidade de ampliação e redução para uma experiência mais dinâmica.
8. Inserir texto informativo para tornar o cenário mais interativo e para aumentar compreensão do usuário.
9. Realizar testes no programa do Sistema Solar para assegurar seu funcionamento adequado, realizando os ajustes necessários na escala, iluminação e movimentos planetários.

## Capítulo

# 5

## Implementação

### 5.1 Introdução

Assegurar a qualidade gráfica é crucial, uma vez que melhora a experiência do utilizador, previne problemas e erros, reforça a segurança, confiabilidade e otimiza o retorno do investimento. Este capítulo retrata de forma fiel o ambiente desenvolvido, dividindo-se da seguinte forma:

- A secção 5.2, Dependências, explana as dependências do projeto;
- A secção 5.3, Detalhes de Implementação, caracteriza e descreve cada ambiente gráfico desenvolvido.

### 5.2 Dependências

```
#include <glad/glad.h> // Gerencia funcoes de
    ponteiros para o OpenGL, utilizado para
    iluminacao.
#include <GLFW/glfw3.h> // Usado para criar janelas
    , contextos e receber inputs, eventos.
#include <glm/glm.hpp> // Biblioteca do OpenGL para
    funcoes matematicas.
#include <iostream> // Implementa recursos de
    entrada/saida baseados em fluxo.
#include <vector> // Biblioteca que contem um tipo
    de array que pode mudar o seu tamanho.
```

```
#include <math.h> // Fornece funcoes e valores
    matematicos.
#include <stb_image.h> // Inclui as funcoes
    necessarias para obter uma imagem e usa-la para
    as texturas.
#include <ft2build.h> // Biblioteca para a escrita
    de palavras no ecrã.
#include <shader_m.h> // Usado para fazer a leitura
    dos ficheiros shaders, criando assim um objeto
    Shader.
#include <camera.h> // Fornece um objeto Camera com
    funcionalidades para a camara navegar no espaco
    .
```

Tabela 5.1 - imports utilizados

## 5.3 Implementações

### 5.3.1 Planetas

Para criar os planetas, utilizou-se uma esfera como base para quase todos os planetas. Em seguida, as texturas dos planetas foram carregadas em variáveis. Quando era necessário criar um planeta específico, obtinha-se a esfera base e aplicava-se a textura do planeta correspondente. Posteriormente, era criada uma matriz de modelo para o planeta. Calculava-se a posição do planeta, trasladava-se a matriz de modelo para essa posição, e o planeta era então rotacionado em torno de seu próprio eixo (eixo Y). A escala da esfera base era ajustada para se adequar ao tamanho do planeta. Por fim, a nova matriz de modelo era passada para os shaders, permitindo que o planeta fosse desenhado na tela.

Para criar os satélites, como no caso da lua, utilizou-se o modelo da Terra como base. Em seguida, aplicava-se uma translação ao redor da Terra para posicionar o satélite em sua órbita.

### 5.3.2 Céu/Skybox

Para criar uma skybox em OpenGL, começamos por criar um cubo gigante e carregar as imagens para as texturas das faces do cubo. Em seguida, criamos uma estrutura de dados para armazenar as informações da skybox, fazemos o

upload dos vértices do cubo para o VBO e fundamos uma estrutura de dados para armazenar a textura da skybox, fazendo também o upload das imagens para a textura da skybox.

Posteriormente, desenvolvemos um programa de shader específico para a skybox. Desenhamos o cubo utilizando o programa de shader da skybox e a textura da skybox. Além disso, atualizamos a transformação de visualização e desenhamos o cubo novamente quando o observador se move, criando assim a ilusão de um espaço infinito.

### 5.3.3 Orbitas dos Planetas

Primeiramente, criamos um vetor com os valores das órbitas que serão armazenados no buffer. Durante o ciclo while, procuramos os valores armazenados no buffer, percorremos os 8 planetas e realizamos a translação da órbita para a posição do planeta. Em seguida, enviamos os dados para os shaders e desenhamos na tela. Posteriormente, para a órbita da Lua, efetuamos a translação de forma a que a Terra seja o centro da órbita, e enviamos os dados para os shaders para desenhar no ecrã.





## **Capítulo**

# 6

## ***Conclusões e Trabalho Futuro***

### **6.1 Conclusões**

Este projeto possibilitou a criação de uma representação em três dimensões do Sistema Solar, caracterizada pelo seu realismo, detalhe e natureza educativa. Ao aplicar técnicas de Computação Gráfica e realizar uma pesquisa aprofundada sobre o Sistema Solar, conseguimos desenvolver modelos 3D precisos para os planetas, para a lua e para o Sol. A implementação de texturas e materiais realistas, juntamente com animações que ilustram os movimentos dos planetas, aprimorando ainda mais a autenticidade e o detalhe das suas representações. A utilização de recursos interativos, como informações sobre cada elemento do sistema solar, desempenhou um papel crucial na transformação do sistema solar numa ferramenta educacional eficaz. Em suma, a representação do sistema solar criada neste projeto revela-se uma excelente ferramenta para enriquecer o conhecimento e a compreensão das pessoas sobre o nosso Sistema Solar.

### **6.2 Trabalho Futuro**

É fundamental ter em mente que o entendimento sobre o Sistema Solar está em constante evolução, e descobertas adicionais podem ser realizadas no futuro. Assim, é possível que surjam mais detalhes e informações que possam ser incorporados na representação desenvolvida deste projeto. Adicionalmente, outras vertentes da Computação Gráfica podem ser exploradas no futuro, como a criação de visualizações em realidade virtual ou aumentada. Em suma, há uma variedade de oportunidades para a realização de futuros projetos relacionados com este tema.

## 6.3 Bibliografia

<https://www.di.ubi.pt/agomes/cg/>

<https://www.opengl.org/>