

Universidade da Beira Interior

Departamento de Informática



**Departamento de
Informática**

Nº 2 - 2020: *αsteroids - Alpha Defense Team*

Elaborado por:

**Rúben Guilherme, nº 41059
Paulo Duarte, nº 41853**

Orientador:

Professor/a Doutor/a Abel Gomes

14 de janeiro de 2021

Conteúdo

Capítulo

1

Introdução

1.1 Motivação

Este projeto foi desenvolvido, no âmbito da disciplina de Computação Gráfica lecionada pelo professor Abel Gomes, com o intuito de desenvolver e aperfeiçoar as técnicas aprendidas ao longo do semestre.

1.2 História do jogo

1. **Quem é o jogador?** – O jogador faz parte da equipa de defesa Alpha da base espacial UPT-2. Esta base está a navegar por uma cintura de asteroides e devido a isso está a ser atacada por naves inimigas. O jogador deve proteger o caminho da base espacial contra os asteroides e eventuais naves inimigas.
2. **Quem é o inimigo?** – O inimigo faz parte dos planetas que ambicionam destruir a nossa presença no Universo. Depois da destruição da Terra alguns planetas decidiram revoltar-se contra a União Planetária da Terra, criada para unir e guiar as várias nações terráqueas. Estes não querem mais um inimigo a conquistar o cosmos e estão dispostos a tudo para eliminar os humanos.
3. **Porquê?** – Alienígenas estão a atacar o planeta Terra com o intuito de dizimar a raça humana. Assim sendo, os terráqueos não têm outra opção senão fugir, uma vez que a tecnologia dos extraterrestres é muito superior à dos humanos.
4. **Objetivo?** – Fugir do planeta Terra atravessando a cintura de asteroides.

5. **Onde?** – Sistema Solar - Cintura de asteróides

Capítulo

2

Tecnologias e Ferramentas Utilizadas

2.1 OpenGL

OpenGL é a API utilizada para o desenvolvimento do jogo Asteroids em 3D.

2.2 GLFW

A biblioteca GLFW é utilizada para que nos seja possível criar e gerir janelas e interagir com o teclado e com o rato.

2.3 GLM

A biblioteca GLM é utilizada para que consigamos estender o nosso conjuntos de funções matemáticas.

2.4 GLAD

O Glad é utilizado para carregar algumas linguagens necessárias para a implementação do programa nomeadamente o GL.

2.5 ASSIMP

A biblioteca ASSIMP é utilizada para importar os nossos objetos 3D.

2.6 FreeType

A biblioteca FreeType é utilizada para que possamos ler fontes de texto e convertê-las de forma a que as possamos utilizar no OpenGL.

2.7 Blender

Utilizamos o Blender para modelar e texturizar os nossos objetos.

Capítulo

3

Desenvolvimento e Implementação

3.1 Gestão do projeto

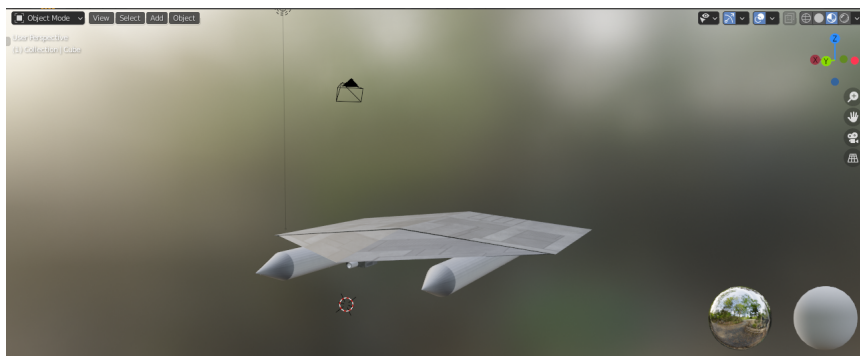
Numa primeira fase criámos um plano de trabalho no qual nos baseámos para a concretização do projeto que queríamos. Começámos por criar e pensar nas funcionalidades que queríamos ver realizadas no nosso jogo. De seguida, estimámos o tempo médio de execução de cada fase do projeto e as respetivas *deadlines*.

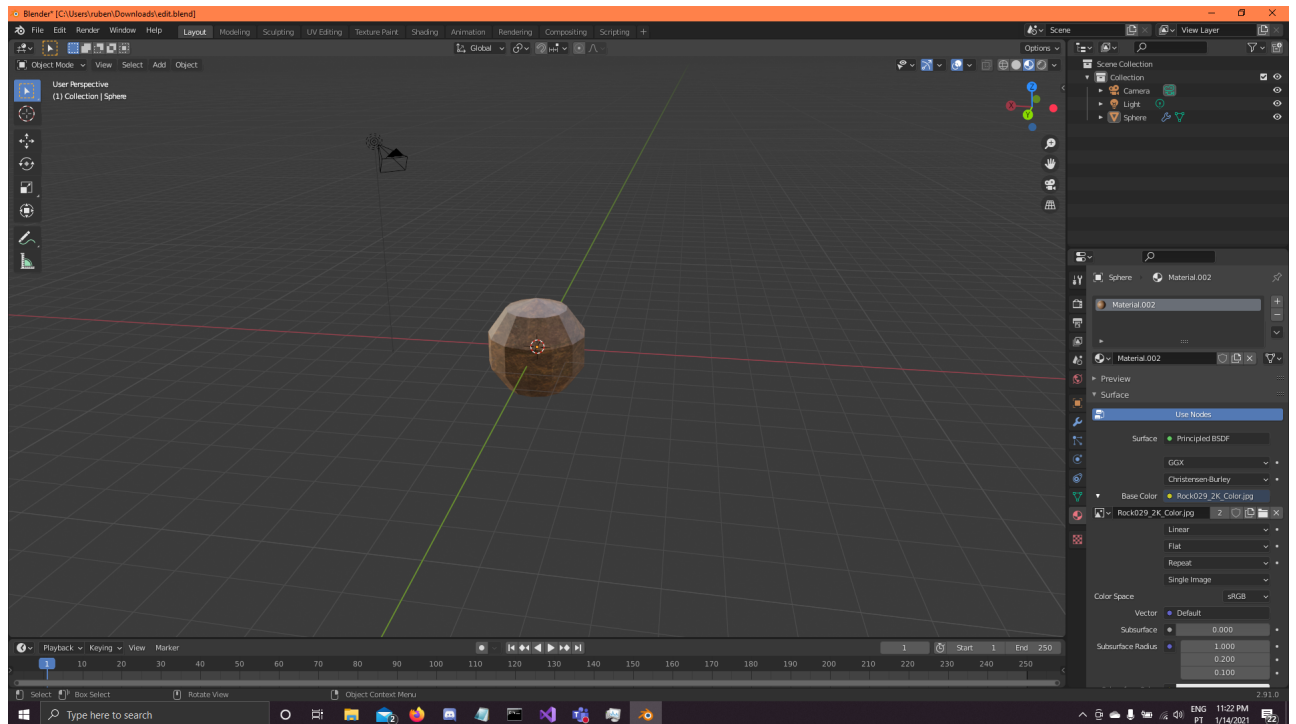
3.2 Parte Técnica

Concluída a fase de gestão do projeto, começámos o seu desenvolvimento.

3.2.1 Modelação

No que diz respeito à fase de **Modelação**, utilizámos o Blender para criar alguns objetos, tais como: a nossa nave, os asteróides e o nosso cenário.





3.2.2 Interação

Na fase de desenvolvimento da **Interação** preocupámo-nos com a escolha de comandos fáceis e intuitivos para manipular a nave. Assim sendo, decidimos utilizar o teclado para a mover, e o rato para controlar a mira da nave e a descarga de plasma para destruir os asteróides.

3.2.3 Shaders

No projeto temos 3 tipos de **Shaders**, um para as texturas, um para o texto e outro para dar load a objetos 2d (como a mira). Estes shaders são baseados naqueles encontrados no website learnopengl.

3.2.4 Texturização

A **Texturização** dos nossos objetos foi realizada conjuntamente com a modelação no Blender.

3.3 Funcionalidades Extra

A equipa de desenvolvimento decidiu implementar algumas funcionalidades extra que achou pertinentes.

3.3.1 Texto Gráfico

Decidimos colocar **Texto Gráfico** no nosso jogo para melhorar a experiência de utilizador.

3.3.2 Sistemas de Pontuação

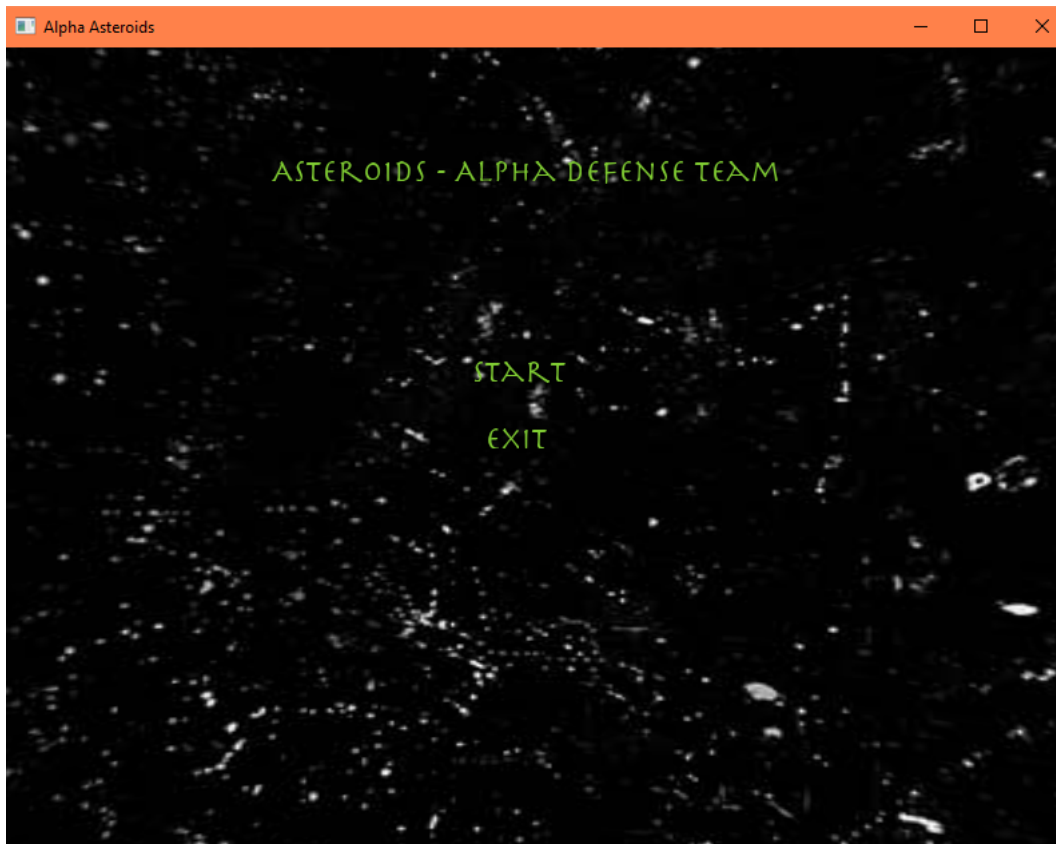
Criámos um **Sistema de Pontuação** para tornar o jogo mais competitivo.

3.3.3 2D -> 3D

Desde a fase de gestão e planeamento do nosso projeto que nunca pensámos em fazer o jogo em 2D dado que queríamos revolucionar o clássico jogo Asteroids com uma versão em 3D.

3.3.4 Menus

Desenvolvemos alguns menus para o nosso jogo, nomeadamente: um menu principal, um menu de pausa e um menu de *Game Over*.



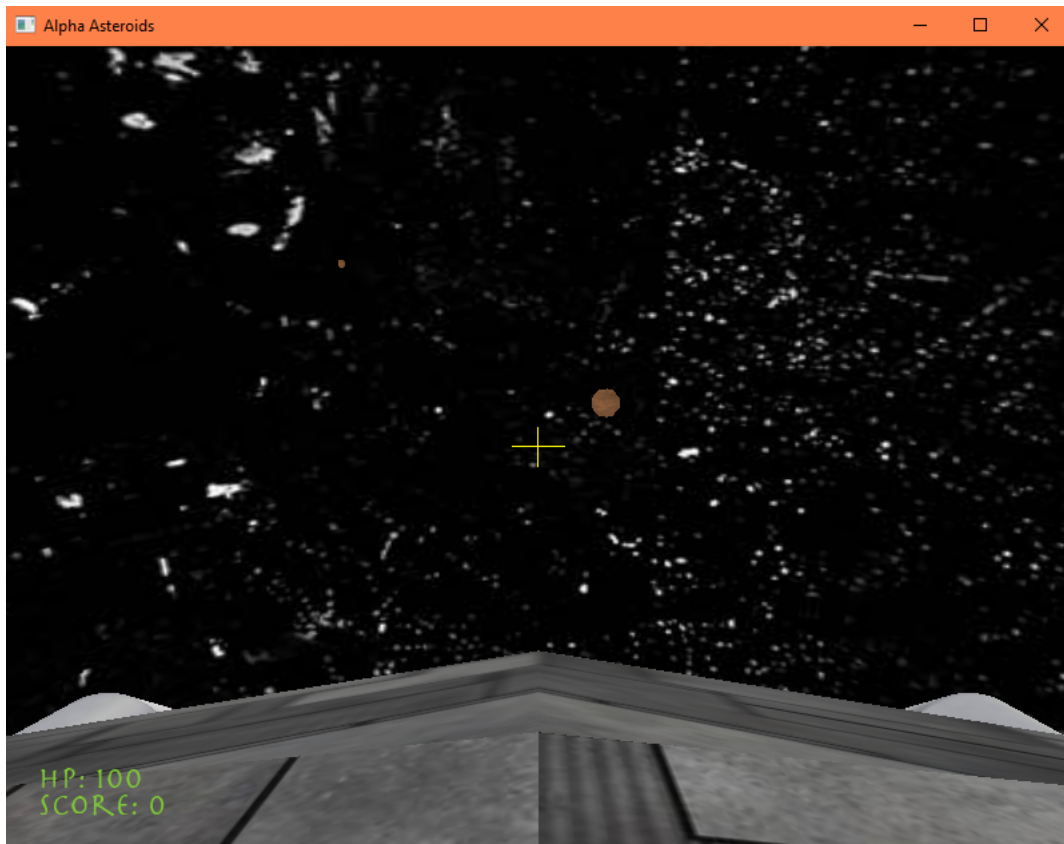
Capítulo

4

Descrição do funcionamento do software

O jogo consiste numa nave que, controlada pelo jogador através do rato e do teclado, tem de destruir os asteróides existentes. Caso a nave não consiga destruir ou desviar-se dos asteróides, esta vai perdendo vida até ser destruída.

A nave movimenta-se num mapa esférico com limites bem definidos, caso a nave se movimente para fora dos limites do mapa, inicialmente aparecerá uma mensagem de aviso que se ignorada pelo jogador, este perderá o jogo tendo de começar de novo.



Capítulo

5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Com a finalização deste projeto pudemos aprender sobre todas as fases de desenvolvimento de uma aplicação gráfica (neste caso de um jogo), assim como desenvolver as nossas capacidades técnicas e soft-skills.

5.2 Trabalho Futuro

A equipa de desenvolvimento acha que o jogo desenvolvido cumpre todos os requisitos estabelecidos, no entanto os programadores tinham mais alguns objetivos que ficaram por concretizar e que podem ser encarados como "trabalho futuro". A equipa gostaria de ter criado mais naves para que o jogador pudesse escolher a nave de que mais gosta, naves inimigas, *bosses*, diferentes cenários e alguns *power-ups*. Além disso, também tínhamos planeado existirem algumas opções de customização nomeadamente ao que diz respeito à nave, à mira e aos tiros.

Capítulo

6

Bibliografía

1. <https://learnopengl.com/In-Practice/2D-Game/Levels>
2. <https://learnopengl.com/Getting-started/Transformations>
3. <https://learnopengl.com/Model-Loading/Assimp>
4. <https://learnopengl.com/Model-Loading/Mesh>
5. <https://learnopengl.com/Model-Loading/Model>
6. <https://learnopengl.com/Getting-started/Camera>
7. <https://github.com/lucatironi/cpp-gl-asteroids>
8. <https://www.cplusplus.com/reference/vector/vector/>
9. https://www.glfw.org/docs/3.3/input_guide.html
10. <https://cc0textures.com/categories>
11. https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_collision_detection
12. <http://www.cplusplus.com/reference/cmath/sqrt/>

