

# StarCom Shooter

## Relatório



Turma 5 Grupo 06

Diogo Ramos (up202207954)

Gabriel Braga (up202207784)

Guilherme Rego (up202207041)

# Índice

1. Instruções de utilização	
1.1. Menu Inicial .....	3
1.2. Definições .....	4
1.3. Instruções.....	4
1.4. Jogo.....	5
1.5. Pausa .....	5
2. Estado do Projeto	
2.1. Tabela dos dispositivos.....	6
2.2. Timer.....	6
2.3. Teclado .....	7
2.4. Rato .....	7
2.5. Placa gráfica .....	7/8
2.6. RTC .....	8
3. Estrutura e Organização do código	
3.1. Proj.....	8
3.2. Video.....	8
3.3. Timer .....	9
3.4. Mouse.....	9
3.5. Keyboard .....	9
3.6. Rtc .....	9
3.7. Date Controller.....	10
3.8. Instructions.....	10
3.9. Sprite .....	10
3.10. Game.....	10
3.11. Difficulty Controller .....	10
3.12. Game Controller .....	11
3.13. Mouse Controller .....	11
3.14. Menu.....	11
3.15. Settings.....	11

3.16.	Pause.....	11
3.17.	Game over.....	12
3.18.	Call Graph.....	13
4.	Detalhes de Implementação.....	14
5.	Conclusão.....	14/15

## Instruções de utilização

### Menu inicial

Depois de correr o programa, é apresentado ao utilizador um menu onde pode escolher entre 4 opções:

- Jogar
- Definições
- Instruções
- Sair



## Definições

Ao aceder ao menu das definições, o utilizador pode escolher apagar a leaderboard do jogo.



## Instruções

Na página de instruções, o utilizador obtém informação de como funciona o jogo, e que teclas usar para conseguir experimentar a gameplay.





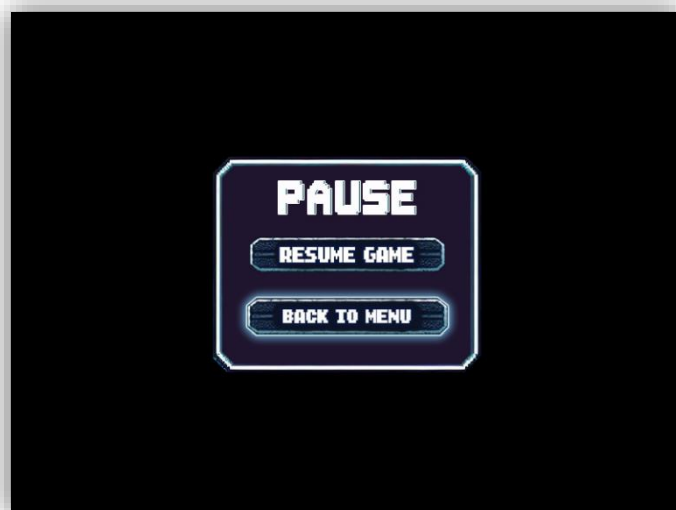
## Jogo

Quando o user escolher jogar, aparece uma nave na posição central do ecrã, e de 8 posições diferentes podem surgir asteroides que se movem em direção à nave, as setas fazem com que o canhão rode à volta e o espaço faz o mesmo canhão disparar para eliminar os asteroides. Quando um asteroide colidir com a nave o utilizador perde e aparece uma tela de game over.



## Pausa

A meio do jogo o utilizador pode precisar de parar de jogar por alguns instantes e ao clicar na tecla esc o jogo fica em pausa, depois o jogador pode escolher se quer sair ou continuar o progresso.



# Estado do Projeto

## Tabela dos dispositivos

Dispositivo	Funcionalidade	Interrupções
Timer	Criação de asteroides e também cooldown de disparo.	Sim
Rato	Navegar entre as opções dos menus, e acertar em powerups.	Sim
Teclado	Movimento do canhão e disparo, e escolha das opções dos menus.	Sim
Placa Gráfica	Toda a parte visual do jogo, desde menus, até ao jogo em si.	Não
RTC	Mostrar ao utilizador a data atual em que está a jogar.	Sim

## Timer

A cada interrupção usa-se o timer para atualizar o estado do jogo. As principais funções, seja para desenhar o menu (`draw_menu()`) ou até mesmo no próprio jogo para passado um determinado tempo criar um novo asteroide com a função (`create_asteroid()`), só são chamadas porque o timer existe, e é fundamental em todo o funcionamento e dinâmica do jogo. O timer é basicamente o “núcleo” do nosso jogo.

## Rato

O rato não é tão usado no nosso projeto, apenas serve para navegar entre as diferentes opções do menu e para obter powerups quando eles aparecem no jogo.

## Teclado

O teclado é um dos devices mais importantes do nosso projeto. No menu ao pressionar as setas para baixo e para cima, o utilizador consegue entre as diferentes opções. No jogo em si é fundamental o uso do teclado uma vez que as setas para a esquerda e para a direita permitem definir a posição do canhão, e a tecla espaço faz com que se o canhão tiver balas dispare para a posição onde está virado. As funções implementadas nos laboratórios servem para avaliar os scancodes das teclas premidas para que o jogo seja funcional e cria toda a dinâmica deste.

## Placa Gráfica

É responsável por todo o visual do jogo, sem a placa gráfica não seria sequer possível criarmos o nosso projeto. No nosso jogo usamos double buffering, que consiste em desenhar tudo o que queremos que o utilizador veja num buffer secundário, e assim que esteja tudo desenhado nesse buffer, faz-se a cópia para o buffer principal, com esta opção evita-se que o utilizador veja o ecrã a “piscar”, as funções principais para o double buffering são `allocate_buffers()` , `deallocate_buffers()` , `swap_buffers()`. Para ver se existem colisões, a placa gráfica também é importante, pois dá para perceber se existem colisões entre xpm's, o que é fulcral no nosso jogo, para perceber se os asteroides são eliminados ou

não, com a função `check_collisions()`. E por fim para mostrar todos os xpm's criados, a placa gráfica permite fazer um jogo com um aspeto visual agradável para o utilizador.

## RTC

O real time clock serve essencialmente para a mostragem de que dia, mês, e hora são num dado momento, para que o utilizador possa ver a hora sem ter de sair do jogo.

# Estrutura e Organização do código

## PROJ

Tem basicamente a base do projeto, onde é configurada a resolução e o modo da placa gráfica, o carregamento dos xpm's e também a chamada à função do jogo, tem também chamadas a funções para a limpeza do buffer e para sair do modo gráfico quando o utilizador sai do jogo.

**Peso no projeto: 10%**

## Video

Foi implementado no lab5 e agora usado no projeto, adicionamos apenas funções para tratar do double buffering do jogo.

**Peso no projeto:15%**



## Timer

Dispositivo configurado no lab2, é o núcleo do nosso jogo, pois a cada interrupção analisa o estado do jogo.

**Peso no projeto:15%**

## Mouse

Foi implementado no lab4 e é o dispositivo menos usado no projeto, serve para navegar e para conseguir obter powerups durante o jogo.

**Peso no projeto: 5%**

## Keyboard

Juntamente com o timer é um dispositivo fulcral no jogo, pois serve para navegar e jogar, foi transportado do lab3 para o projeto.

**Peso no projeto:15%**

## RTC

Implementação de um device não implementado durante os labs e que nos permite saber a hora e a data atual em que o jogo está a correr.

**Peso no projeto:5%**

## Date Controller

Funções para visualizar a hora e a data atual do jogo.

**Peso no projeto:2%**

## Instructions

Mostra o menu das instruções, que o utilizador pode ver para aprender a jogar.

**Peso no projeto:2%**

## Sprite

Contem funções importantes para desenhar menus, asteroides, score e também a nave principal do jogo.

**Peso no projeto:5%**

## Game

Contem o as funções para analisar em que estado se encontra o jogo e para interpretar os scancodes do teclado, e chama as funções para desenhar os diferentes xpm's mediante o ponto onde se encontra o jogo.

**Peso no projeto:10%**

## Difficulty Controller

Serve para analisar a pontuação do jogador e incrementar a dificuldade à medida que a pontuação vai aumentando.

**Peso no projeto: 2%**

## Game Controller

Tem as funções principais do jogo, como detetar colisões para aumentar o score, e também desenho dos xpm's do canhão, dos asteroides e também dos lasers que vão destruir os asteroides.

**Peso no projeto: 5%**

## Mouse Controller

Contem a função para desenhar o cursor do rato.

**Peso no projeto: 2%**

## Menu

Responsável por desenhar os principais menus do jogo.

**Peso no projeto: 2%**

## Settings

É destinado a desenhar as definições do jogo, como por exemplo,limpar a leaderboard.

**Peso no projeto:1%**

## Pause

Responsável pelo desenho do menu de pause.

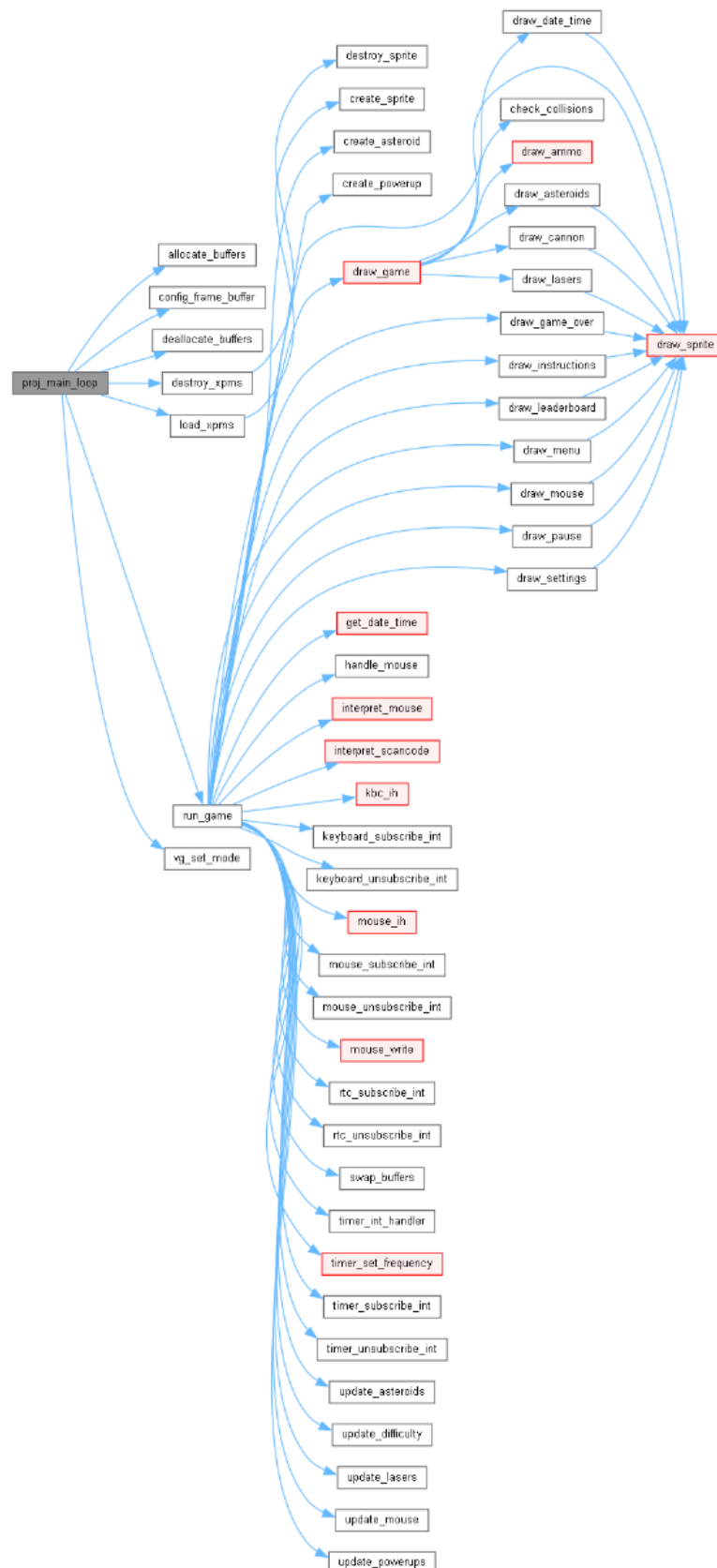
**Peso no projeto:2%**

## Game Over

Contém a função para desenhar a tela quando o utilizador perde o jogo.

**Peso no projeto:2%**

# Call Graph



# Detalhes de implementação

Uma das principais implementações para que o jogo fique visualmente agradável é sem dúvida nenhuma a técnica de “double buffering”, pois no início do projeto o jogo estava sempre a “piscar”, portanto é sem dúvida muito importante para que o jogo esteja agradável de se jogar.

Outro aspeto importante é a struct que tem os estados do jogo, pois é trivial para o programa perceber quando é que tem de desenhar cada aspeto do jogo, desde os diferentes menus até ao jogo em si no desenho do canhão, asteroides e lasers.

Por fim, algo que tivemos de aprender a como implementar foram as colisões do jogo, que é bastante importante para perceber se o laser embateu no asteroide, para que assim o utilizador possa somar pontuação.

## Conclusão

Os problemas foram mais no início, pois era preciso definir todo o projeto e tudo o que queríamos fazer, e claro tínhamos de aprender a trabalhar todos em equipa e com eficiência para que não demorássemos o dobro do tempo.

Algo que se tivéssemos mais tempo podíamos explorar era introdução de mais powerups no jogo, talvez um modo 1vs1, e mais modo de resolução para que o utilizador pudesse escolher.



O projeto foi algo que nos motivou mais um bocado para a cadeira, uma vez que, é uma cadeira que para perceber a fundo é necessário muito tempo, e com todos os outros projetos e cadeira é complicado de conciliar, mas com este jogo criado conseguimos perceber melhor como funciona o minix e todos os dispositivos implementados durante os laboratórios.

A principal lição é que com um trabalho gradual e semana a semana , conseguimos obter um projeto bom e com um aspeto agradável para o utilizador que possa vir a experimentar.