|  |
| --- |
|  |
| *Space Invaders* |
|  |
| Laboratório de Computadores - 2015/2016 |

**LCOM1516-T6G12**

Bernardo Ferreira dos Santos Aroso Belchior – up201405381

Edgar de Lemos Passos – up201404131

4 de janeiro de 2016

Space Invaders

Laboratório de Computadores - 2015/2016

Conteúdo

[1. Instruções de utilização 3](#_Toc439666969)

[1.1. Menu inicial 3](#_Toc439666970)

[*1.2.* *Singleplayer* 4](#_Toc439666971)

[*1.3.* *Multiplayer* 6](#_Toc439666972)

[*1.4.* *Highscores* 7](#_Toc439666973)

[7](#_Toc439666974)

[1.5. *Options* 8](#_Toc439666975)

[*1.6.* *Game Over Menu* 9](#_Toc439666976)

[2. Estado do projeto 10](#_Toc439666977)

[2.1. Dispositivos utilizados 10](#_Toc439666978)

[*2.2.* *Timer* 11](#_Toc439666979)

[2.3. Teclado 12](#_Toc439666980)

[2.4. Rato 13](#_Toc439666981)

[2.5. Placa Gráfica 14](#_Toc439666982)

[*2.6.* *Real Time Clock* 15](#_Toc439666983)

[*2.7.* *Serial Port* 16](#_Toc439666984)

[3. Organização e estrutura do código 17](#_Toc439666985)

[*3.1.* *Aliens* 17](#_Toc439666986)

[*3.2.* *Animation* 17](#_Toc439666987)

[*3.3.* *Bitmap* 17](#_Toc439666988)

[*3.4.* *Button* 17](#_Toc439666989)

[*3.5.* *Events* 17](#_Toc439666990)

[3.6. FIFO 18](#_Toc439666991)

[*3.7.* *Font* 18](#_Toc439666992)

[*3.8.* *Game Over Menu* 18](#_Toc439666993)

[*3.9.* *Highscore* 18](#_Toc439666994)

[*3.10.* *I8042* 18](#_Toc439666995)

[*3.11.* *I8254* 18](#_Toc439666996)

[*3.12.* *Keyboard* 18](#_Toc439666997)

[*3.13.* *Menu* 19](#_Toc439666998)

[*3.14.* *Mouse* 19](#_Toc439666999)

[*3.15.* *Options* 19](#_Toc439667000)

[*3.16.* *Player* 19](#_Toc439667001)

[*3.17.* *Proj* 19](#_Toc439667002)

[*3.18.* *Projectile* 19](#_Toc439667003)

[*3.19.* *RTC* 20](#_Toc439667004)

[*3.20.* *Shield* 20](#_Toc439667005)

[*3.21.* *Singleplayer* 20](#_Toc439667006)

[*3.22.* *Start Menu* 20](#_Toc439667007)

[*3.23.* *State* 20](#_Toc439667008)

[*3.24.* *Timer* 20](#_Toc439667009)

[3.25. UART 21](#_Toc439667010)

[*3.26.* *VBE* 21](#_Toc439667011)

[*3.27.* *Versus Multiplayer* 21](#_Toc439667012)

[*3.28.* *Video Gr* 21](#_Toc439667013)

[4. Detalhes da implementação 23](#_Toc439667014)

[5. Conclusões 24](#_Toc439667015)

[Apêndice 25](#_Toc439667016)

# Instruções de utilização

## Menu inicial

Menu inicial do projeto, que contém os submenus principais do jogo:

* *Singleplayer*;



Figura - Menu inicial

* *Multiplayer*;
* *Highscores*;
* *Options*.

A opção *Exit* permite sair do serviço *proj* e voltar ao ambiente normal de consola do *Minix*.

O menu permite o uso do rato para clicar nos diferentes botões, ou então as teclas para cima/baixo para selecionar as diferentes opções, usando a tecla “Enter” para executar a ação subjacente à opção escolhido. Ao pressionar a tecla “Esc”, nenhum botão será selecionado.

## *Singleplayer*

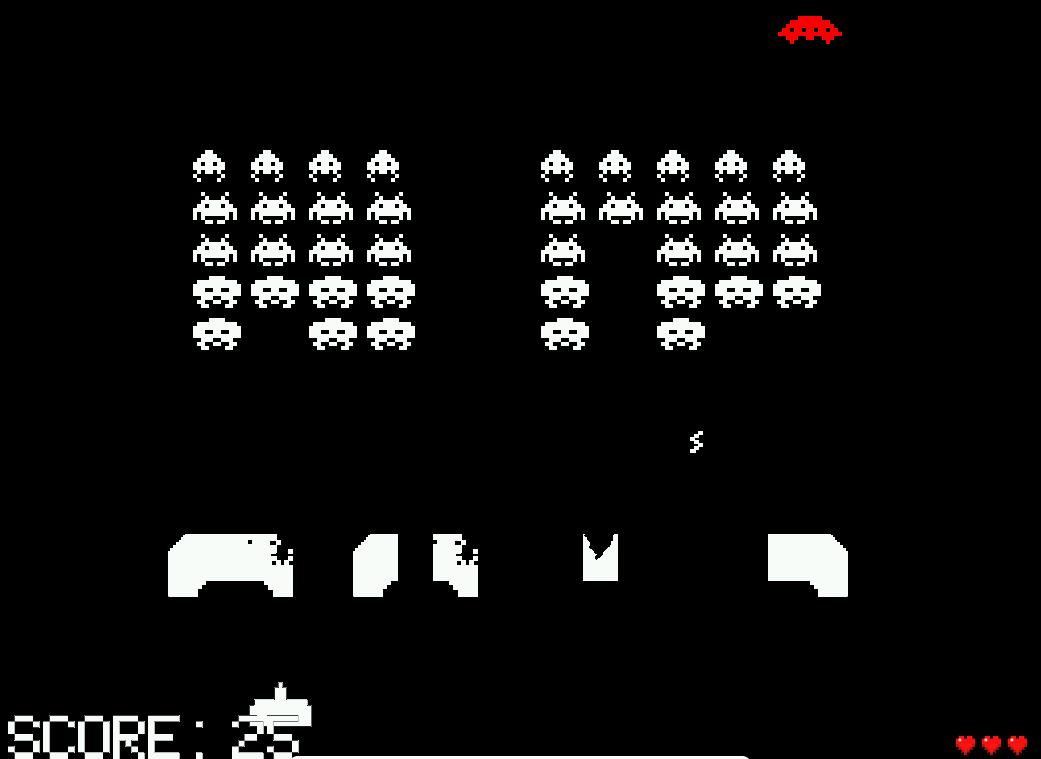


Figura - *Singleplayer*

Acima presenciamos o modo de um só jogador.

O objetivo do jogo é exterminar todos os tipos de extraterrestres presentes na figura 2, estes são as imagens a branco no centro do ecrã. Caso o jogador não o consiga concretizar antes dos invasores atingirem os escudos, o jogador perde. Outro critério de derrota é o número de vidas, representadas no canto inferior direito. O utilizador tem, por defeito, três vidas, e perde uma cada vez que é atingido. Quando o jogador tiver apenas uma vida e chocar contra um projétil, perde o jogo.

Quando morto, cada tipo de extraterrestre fornece uma quantia diferente de pontos ao utilizador, contabilizada no canto inferior esquerdo do ecrã (*score*), que é a medida de comparação entre diferentes jogos.

O jogador também tem a seu dispor um conjunto de quatro escudos que pode utilizar para se proteger dos projéteis lançados pelos invasores.

No topo do ecrã, aleatoriamente, aparecerá uma nave a vermelho que fornece pontos extra quando é atingida.

É também possível sair do jogo pressionando a tecla “Esc”.

Saindo do modo *singleplayer* obrigará o programa a decidir se a pontuação do jogador é alta o suficiente para entrar na tabela das melhores pontuações. Caso o resultado seja afirmativo, será apresentado ao jogador o “Game Over Menu” (ver pág. 9), caso contrário, este será levado para o menu inicial.

## *Multiplayer*



Figura - *Multiplayer*

Acima apresenta-se o modo multijogador que consiste em dois jogadores que disparam projéteis um contra um outro. Quando um deles é atingido, a pontuação do adversário é incrementada. Ao fim do tempo mostrado no canto superior direito é apurado o vencedor. Este será o que tiver mais pontos. Esta pontuação não será adicionada à tabela das melhores pontuações.

## *Highscores*

## 

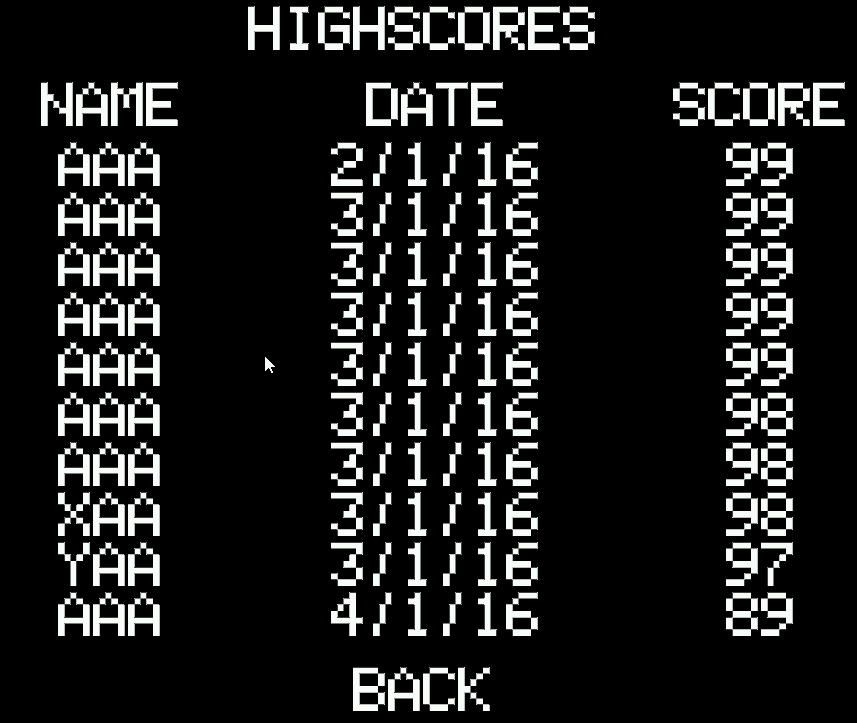


Figura - *Highscores*

Do lado direito está representado do menu das melhores pontuações com o nome (3 letras, ver “Game Over Menu”), data (dia/mês/ano) e pontuação respetivas.

O menu permite o uso do cursor para clicar no botão “Back” para voltar ao menu inicial ou o uso da tecla “Esc”.

## *Options*

Abaixo está representado o menu de opções em que é possível escolher o tipo de controlador a utilizar no modo de um jogador. As opções são: *keyboard*, que permite usar as teclas para a direita e esquerda e barra de espaços do teclado para movimentar a nave do jogador; e *mouse* que utiliza o movimento do rato e o botão do lado esquerdo para mover e disparar, respetivamente.

O menu é passível de ser navegado utilizando as teclas para a direita e esquerda para alterar o controlador a usar, assim como as teclas “Esc” e “Enter” para cancelar e aceitar as alterações feitas. É também possível utilizador o rato para estas ações.



Figura - *Options*

## *Game Over Menu*

Caso a pontuação do jogador no modo *singleplayer* seja suficiente para figurar na tabela das melhores pontuações, este será presentado com o seguinte menu.



Figura - *Game Over Menu*

O menu de fim de jogo permite ao utilizador introduzir um nome identificativo de três letras que, mais tarde, será adicionado à tabela das melhores pontuações.

Este menu poderá ser navegado utilizado o rato.

# Estado do projeto

## Dispositivos utilizados

|  |  |  |
| --- | --- | --- |
| Dispositivo | Utilização | Interrupção? |
| Timer | Atualizações do estado do jogo | Sim |
| Teclado | Interface do jogo com o utilizador e navegação nos menus | Sim |
| Rato | Interface do jogo com o utilizador e navegação nos menus | Sim |
| Placa Gráfica (modo vídeo) | Desenho de imagens | Não |
| Real Time Clock | Obtenção de data das melhores pontuações | Não |
| Serial Port | Comunição entre dois computador para o modo multijogador | Sim |

## *Timer*

O *timer* presente no *Minix* é utilizado no sentido de atualizar o estado do programa. Quer seja necessário desenhar imagens, mover extraterrestres e disparar projéteis. Praticamente todo o tipo de informação é modificado pelas interrupções do temporizador, sendo este indispensável para o bom funcionamento de todo o projeto.

## Teclado

O teclado, sendo um dispositivo não tão importante para o sucesso do trabalho, também tem a sua fatia de atenção. Os jogadores quer no modo de apenas um jogador, quer em multijogador têm a opção de utilizar as teclas para controlar as suas naves. Contudo, não é só nos modos de jogo que este dispositivo é usado. Na interação jogador-menu, o teclado tem um papel alternativo, mas simultâneo, ao do rato, alargando as opções de escolha para o utilizador.

## Rato

Este periférico mantém uma relação chegada ao utilizador, uma vez que é o modo mais utilizado de navegação nos menus. No entanto, também poderá ser utilizado no modo de um jogador para movimentar a nave e lançar projéteis.

## Placa Gráfica

A placa gráfica é um componente chave do projeto, dado que permite desenhar imagens criadas pelo grupo, utilizando o modo RGB (5:6:5) com 1024 pixéis de largura por 768 de altura.

O grupo decidiu utilizar a técnica de *double buffering* de modo a fornecer uma experiência mais suave ao jogador.

De modo a detetar colisões entre entidades do jogo, o grupo desenvolveu um algoritmo de deteção de colisões entre retângulos.

Numa tentativa de embelezar o ambiente de jogo, foram criadas animações utilizando *bitmaps*.

Para ser possível o envio de informação textual, o grupo criou um algoritmo para escrever inteiros, caracteres e *strings* através de um *bitmap* baseado na tabela ASCII.

## *Real Time Clock*

O relógio em tempo real é utilizado no projeto com o intuito de ler a data de obtenção de uma das melhoras pontuações. Como tal, o relógio só é acedido quando necessário, não sendo necessário gerar interrupções.

## *Serial Port*

Parcialmente implementada nos módulos FIFO e UART. O grupo não conseguiu desenvolver a infraestrutura necessária para o bom funcionamento da porta de série a tempo da entrega.

# Organização e estrutura do código

## *Aliens*

Lida com todo o tipo de interações jogador-extraterrestres.

Este módulo foi desenvolvido maioritariamente pelo elemento Edgar Passos (80%).

## *Animation*

Contém as sprite de cada animação, permite criar animações a partir de nomes de ficheiros, ou de bitmaps já carregados.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Bitmap*

Carrega, desenha e apaga *bitmaps.* Código publicado por Henrique Ferrolho ([origem](http://forums.fedoraforum.org/showthread.php?t=171389)), no seu [blog](http://difusal.blogspot.pt/2014/09/minixtutorial-8-loading-bmp-images.html), com variadas modificações, nomeadamente em termos de carregamento de imagens. As imagens em formato *bmp* são guardadas pelo programa de edição de imagens “Gimp 2” com as linhas invertidas verticalmente. Enquanto o código original lê a imagem como está e a desenha invertida, o código alterado ao carregar a imagem já a espelha segundo um eixo horizontal, e portanto, encontra-se em memória como deverá ser impressa.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Button*

Cria, lida com colisões, desenha e destrói botões. É permitido criar botões com *bitmaps* como fundo e/ou texto.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Events*

Lida com os eventos lançados pelas interrupções, quer estes sejam cliques do rato, interrupções do *timer* ou interação do utilizador do teclado.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## FIFO

Contém a implementação de uma fila em C, para utilização com a *serial port*.

Este módulo foi desenvolvido unicamente pelo elemento Edgar Passos.

## *Font*

A partir de um *bitmap*, com um formato específico, que contém caracteres ordenados pelo seu código ASCII, desenha inteiros, caracteres e *strings*.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Game Over Menu*

Menu criado quando um jogador finaliza o jogo de único jogador e a sua pontuação é suficiente para entrar na tabela das melhores pontuações.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Highscore*

“Classe” que contém a informação das dez melhores pontuações obtidas no modo de um jogador. Lê e escreve para um ficheiro “highscore.txt”.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *I8042*

Importado do código das aulas práticas do laboratório 3.

## *I8254*

Importado do código das aulas práticas do laboratório 2.

## *Keyboard*

Importado do código das aulas práticas do laboratório 3.

## *Menu*

Contém todos os botões pertencentes a cada menu e trata do seu desenho e colisão.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Mouse*

Importado do código das aulas práticas do laboratório 4.

## *Options*

Menu de opções. Contém todas as interações do utilizador com o menu.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Player*

Contém toda a informação sobre um jogador e trata da sua colisão com projéteis.

Este módulo foi desenvolvido unicamente pelo elemento Edgar Passos.

## *Proj*

Módulo principal: inicializa variáveis, subscreve interrupções e inicializa o modo gráfico. Contém o ciclo principal do jogo. Após isso, apaga as variáveis, sai do modo gráfico e faz *unsubscribe* às interrupções subscritas.

Este módulo foi desenvolvido igualmente pelos elementos do grupo (50%/50%).

## *Projectile*

Trata da criação, desenho e destruição dos projéteis. Implementa uma lista duplamente ligada.

Este módulo foi desenvolvido unicamente pelo elemento Edgar Passos.

## *RTC*

Lê informação contida nos registos do *Real Time Clock.*

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Shield*

Contém toda a informação relativa aos escudos do jogo.

Este módulo foi desenvolvido unicamente pelo elemento Edgar Passos.

## *Singleplayer*

Trata todos os acontecimentos relativos ao modo de um só jogador.

Este módulo foi desenvolvido igualmente pelos elementos do grupo (50%/50%).

## *Start Menu*

Menu inicial e tudo o que lhe está subjacente.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *State*

Contém todos os estados e lida com as suas transições, eliminado o estado anterior, antes de evoluir para o seguinte.

Este módulo foi desenvolvido unicamente pelo elemento Bernardo Belchior.

## *Timer*

Importado do código das aulas práticas do laboratório 2.

## UART

Módulo contendo o código relativo à porta de série.

Este módulo foi desenvolvido unicamente pelo elemento Edgar Passos.

## *VBE*

Importado do código das aulas práticas do laboratório 5.

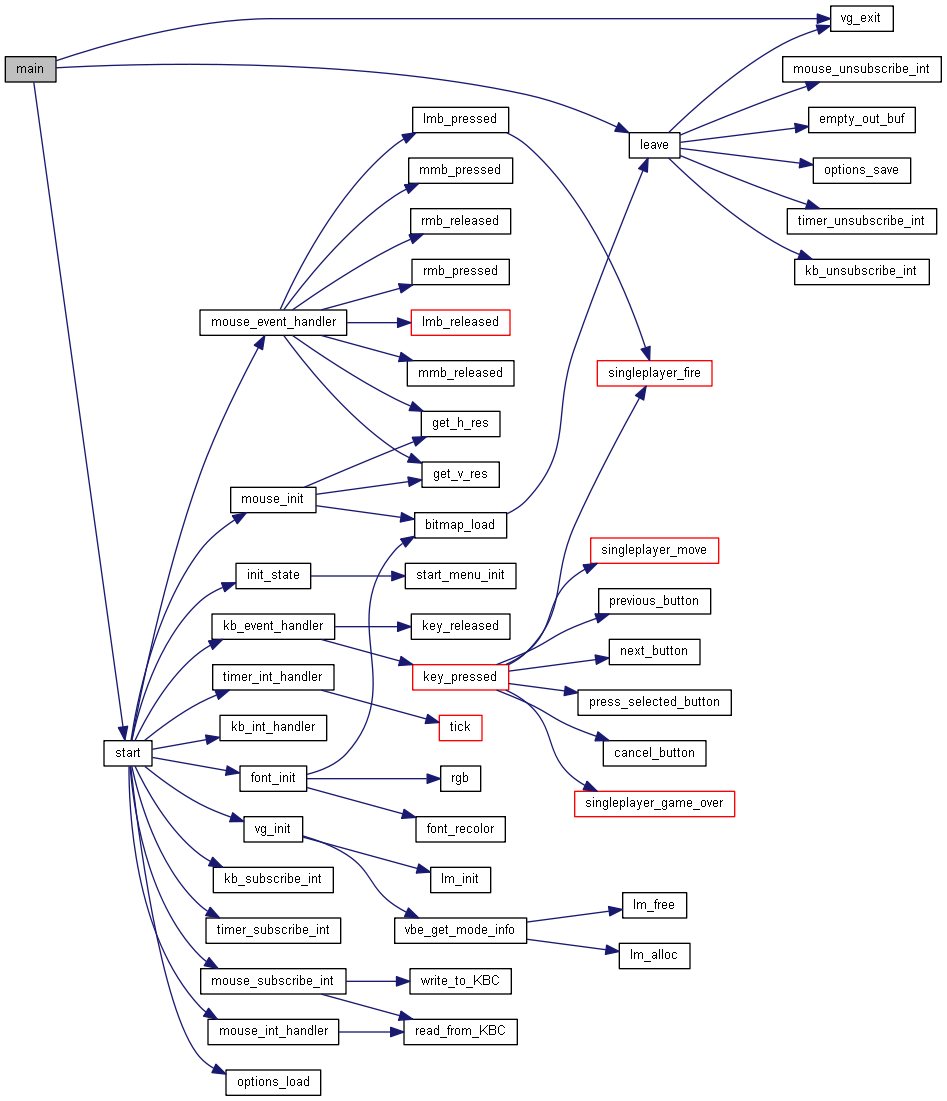
## *Versus Multiplayer*

Trata todos os acontecimentos relativos ao modo multijogador.

Este módulo foi desenvolvido unicamente pelo elemento Edgar Passos.

## *Video Gr*

Importado do código das aulas práticas do laboratório 5.

* 1. **Call Graph**

# Detalhes da implementação

O grupo utilizou código por camadas de modo a facilitar a utilização e escrita de funções mais complexas. É exemplo o módulo “font”, que permite escrever texto de forma muito simples, apesar da complexidade que se encontra por detrás do código. A facilidade de criar menus e botões é mais uma prova da boa estrutura construída pelo grupo.

O módulo *bitmap*, apesar de ter sido inicialmente copiado (ver módulo *Bitmap*) foi muito modificado de modo a adaptar-se às necessidades do grupo.

# Conclusões

O grupo sente que se esforçou e dedicou muitas horas a este projeto, contudo, a falta de informação em certos aspetos são menos oportunos.

O elemento Bernardo Belchior utilizou muito do seu tempo para a pesquisa acerca da chamada de funções em *assembly* a partir de código em C. Por sua vez, o elemento Edgar Passos denotou muita dificuldade na implementação da porta de série que acabou por não ser finalizada.

# Apêndice

Devido à utilização de imagens e da necessidade de utilizador caminhos consistentes, o programa inclui algumas diferenças em relação ao convencional *make*.

Instruções de instalação:

* Entrar no diretório “scripts”
* Correr o *script* “install.sh”
* Correr o *script* “compile.sh”
* Correr o *script* “run.sh”