# 

# Mestrado Integrado em Engenharia Informática e Computação

“Space Invaders”

# Laboratório de Computadores 2019/2020

## Turma 2 Grupo 5



Diogo Reis, up201405015

Miguel Gomes, up201605908

Índice

Introdução…...………………………………………………….………………………3

Como usar este projeto…………………………….………………….………………4

Periféricos Implementados……………………………………………………………6

Organização e estrutura do código…………………………………………………..7

Gráfico de chamada das funções do projeto………………………………………..8

Conclusão………………………………………...…………………………………….9

Introdução

Na unidade curricular de Laboratório de Computadores foi solicitada a realização de um projeto que integrasse a maior parte dos periféricos e técnicas lecionadas. Foi decidido desenvolver um jogo, porque é facilmente possível aplicar toda a matéria abordada ao longo do semestre.

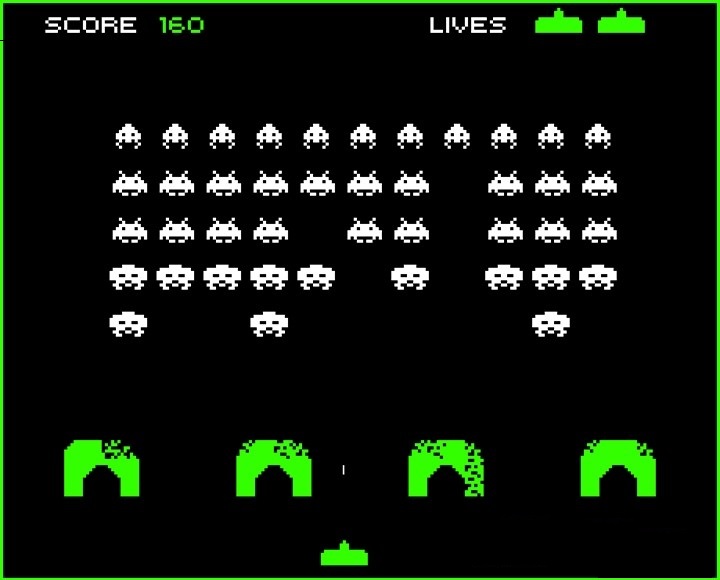
Optamos por desenvolver um jogo semelhante ao “Space Invaders”, um famoso jogo de “arcade” lançado originalmente em junho de 1978 no Japão. Neste jogo o utilizador controla uma nave e tenta repelir uma invasão de naves alienígenas. Quando o jogador consegue abater uma nave inimiga ganha pontos. O objetivo de jogo é conseguir o máximo de pontuação possível.

Figura 1: "Space Invaders" original lançado em 1978

O jogo desenvolvido não é uma recriação fiel do “Space Invaders” original, apresentado algumas alterações à jogabilidade e dificuldade. A alteração mais considerável é que a nuvem de inimigos está sempre a avançar em direção à nave do jogador. Isto força o jogador a ter de eliminar os inimigos todos de cada nível num certo intervalo de tempo, arriscando-se a que as naves inimigas inevitavelmente colidam com a nave do utilizador. O número de tiros que as naves inimigas disparam também é superior ao jogo original.

Como usar este projeto

O projeto não possui nenhum argumento inicial. Basta executar o comando “lcom\_run proj” no MINIX para iniciar o jogo. Para compilar apenas é necessário executar o comando “make”, que utiliza o “Makefile” desenvolvido. Quando o utilizador executa este comando será apresentado no ecrã o menu principal do jogo.

Figura 2: Menu inicial do projeto

Para jogar, basta premir a tecla P no teclado, como é apresentado do menu. Ao pressionar esta tecla o jogador é lançado para o primeiro nível do jogo. Para avançar para os níveis seguintes é necessário eliminar todos os inimigos. Quando o jogador passa de nível ganha um ponto de vida. A nave do utilizador possui munição ilimitada. A pontuação do utilizador é mostrada em tempo real no canto superior esquerdo do jogo.

A nave é controlada de dois modos. Com o teclado, a tecla de espaço faz a nave do utilizador disparar um laser, tecla A faz com que a nave se movimente para a esquerda e tecla D executa o movimento para a direita. Alternativamente podem ser usados os botões do rato para controlar o movimento da nave. O botão do rato esquerdo faz a nave movimentar-se para a esquerda e o botão do rato direito faz a nave movimentar-se para a direita.

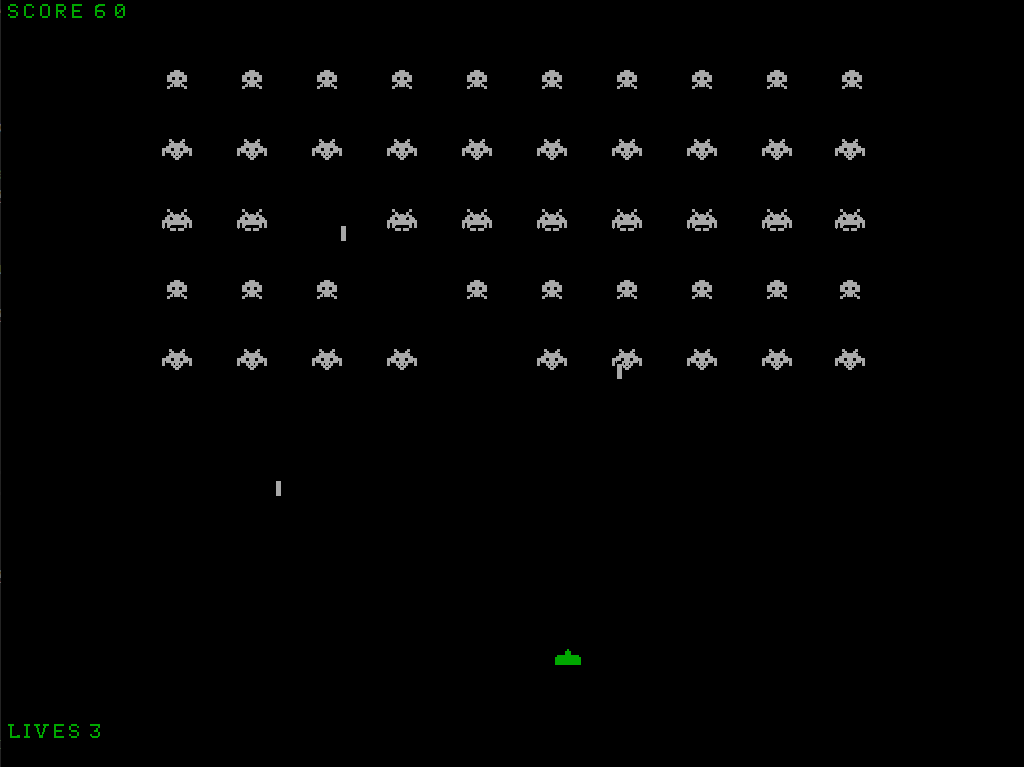
Quando o jogador for atingido perde um ponto de vida. Os pontos de vida são mostrados no ecrã. Quando não tiver mais pontos de vida o jogo acaba e é apresentado um ecrã com a pontuação que obteve nessa partida.

Figura 3: Ecrã principal do jogo

Para voltar ao menu inicial basta premir a tecla M. Caso o utilizador pretenda sair do jogo, basta pressionar a tecla ESC em qualquer ecrã. Caso volte ao menu a meio do jogo ou carregue na tecla ESC, a pontuação é sempre guardada na base de dados.

Figura 4: Ecrã de GameOver

Periféricos implementados

|  |  |  |
| --- | --- | --- |
| **Periférico** | **Funcionalidade** | **Interrupções** |
| Timer | Grafismo do jogo | Sim |
| Teclado | Movimento da nave e navegação nos menus | Sim |
| Rato | Movimento da nave | Sim |
| Placa Gráfica | Grafismo do jogo | Não |
| RTC | Data e hora da pontuação | Não |

Abaixo são apresentados mais detalhes sobre a utilização de cada periférico que foi implementado.

* **Timer e Placa Gráfica:** Foi utilizado o temporizador em conjunto com a placa gráfica para tratar de toda a componente gráfica. Foi utilizado um “buffer” adicional, designado por “double\_buffer” no projeto, para que a cada interrupção do temporizador só ser necessário escrever uma vez na memória de vídeo. Isto corrigiu a existência de “flickering”. Estes dois periféricos são os dispositivos mais importantes do jogo.
* **Teclado:** Controlar o movimento da nave e navegar pelos vários menus. Quando o utilizador carrega numa tecla é gerada uma interrupção que chama uma função responsável por reconhecer a tecla pressionada e tratar a informação devidamente.
* **Rato:** Controlar o movimento da nave. Mecanismo implementado igual ao teclado.
* **RTC:** Armazenar o dia e a hora em que o utilizador obteve uma determinada pontuação, assim como mostrar em que dia e hora foi obtida as melhores pontuações no ecrã de GameOver.

Organização e estrutura do código

Todos os ficheiros que criamos para realizar este projeto assim como a sua função. Todos estes ficheiros foram desenvolvidos por todos os membros do grupo, ou seja, cada elemento realizou 50% de todos os módulos.

* **Proj.c:** Este ficheiro é responsável pelo “loop” principal do menu inicial.
* **Screens.c:** Neste ficheiro são executados os “loops” que dão origem aos vários ecrãs de jogo assim como ao ecrã de “gameover”.
* **Game.c:** Toda a lógica do jogo. “Structs” para representar o que é uma nave, o que é um laser e o que é um inimigo.
* **Timer.c, Keyboard.c, Mouse.c e VBE.c:** Funções desenvolvidas durante as aulas laboratoriais que tratam os respetivos periféricos.
* **RTC.c:** Funções que obtêm a data e hora do computador recorrendo ao “Real Time Clock” do computador.
* **db.c:** Funções que armazenam as pontuações, assim como a data e hora em que foram obtidas numa base de dados. A base de dados que utilizamos é o ficheiro “scores.txt” localizado na pasta db.

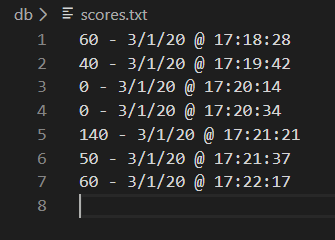


Figura 5: Formato da base de dados

* **Sprite.h:** Informação dos vários “sprites” que usamos para representar formas e letras. A maior parte foi obtida usando o programa GIMP para converter imagens para a extensão “.xpm”.

Todo o código foi comentado de maneira a ser reconhecido pelo “doxygen”. Para mais detalhes pode ser consultado o documento em HTML gerado pelo “doxygen” localizado em “doc/Doxyfile”.

Uma imagem com mapa

Descrição gerada automaticamenteGráfico de chamada das funções do projeto

Conclusão

A realização deste jogo permitiu aplicar toda a matéria lecionada ao longo do semestre ao mesmo tempo. Para além da matéria lecionada também procuramos implementar mais periféricos e técnicas, como foi o caso do “double buffer” e do RTC. Apesar destes tópicos não terem sido abordados nas aulas laboratoriais com a ajuda do professor Rui Grandão Rocha, responsável pelas aulas laboratoriais da nossa turma, facilmente percebemos e implementamos estas funcionalidades.

O rato foi um periférico que não foi explorado como gostaríamos. Gostaríamos de suportar o uso de gestos para movimentar a nave e também implementar um cursor para a navegação pelos menus ser mais intuitiva.

Todos os periféricos e técnicas restantes foram corretamente utilizados e idealizados neste projeto.