

Licenciatura em Engenharia Informática e Computação



Space Invaders

Laboratório de Computadores 2024

2º Ano - 2º Semestre

Turma 14 - Grupo 4

Trabalho Realizado por:

Cláudia Sofia Teixeira Mesquita, up202206173@fe.up.pt

Luana Filipa de Matos Lima, up202206845@fe.up.pt

Pedro Meireles Fonseca Costa, up202206180@fe.up.pt

Sofia Alves Reis, up201905450@fc.up.pt

Índice

1. Introdução	3
2. Instruções de Utilização	4
2.1 Menu Principal	4
2.1 Menu de Modos	5
2.2 Modo de Jogo	6
2.3 Menu Vitória/Derrota	7
3. Estado do Projeto	9
3.1 Funcionalidades	9
3.2 Dispositivos	9
3.2.1 Timer	10
3.2.2 Keyboard	10
3.2.3 Mouse	10
3.2.4 Video Card	10
3.2.5 RTC	10
4. Organização do Código / Estrutura	11
4.1 Timer Module - %	11
4.1.1 alien.c	11
4.1.2 bg.c	11
4.1.3 buttons.c	11
4.1.4 cursor.c	11
4.1.5 letters.c	11
4.1.6 player.c	11
4.1.7 projectiles.c	11
4.1.8 score.c	12
4.1.9 sprites.c	12
5. Detalhes de Implementação	13
Código Dirigido por Eventos	13
Máquinas de Estado	13
Geração de Frames	13
Detecção de Colisões	13
6. Conclusões	14

1. Introdução

A elaboração deste projeto surge no âmbito da unidade curricular Laboratório de Computadores (L.EIC018), desenvolvida pelos alunos da turma 14, grupo 4, do 2º ano da Licenciatura em Engenharia Informática e Computação pela Faculdade de Engenharia da Universidade do Porto.

O nosso projeto procura ser uma recriação do jogo Space Invaders original, onde o jogador é representado por uma nave espacial e o seu objetivo principal é destruir com projéteis os aliens que vão aparecendo no topo da tela. O jogador ganha se conseguir sobreviver durante uma certa quantidade de tempo (dependendo da dificuldade), sendo que para isso deve sobreviver aos tiros dos aliens, enquanto tenta matar o maior número possível destes, antes que lhe toquem.

2. Instruções de Utilização

2.1 Menu Principal

Ao iniciar o programa, o utilizador é direccionado para o menu principal onde poderá usar o rato ou as setas do teclado para seleccionar uma das duas opções, jogar (start) ou sair do jogo (quit):



Fig.1 - Menu Principal

2.1 Menu de Modos

Ao clicar em start, o utilizador é direcionado para o menu de modos onde poderá usar o rato ou as setas do teclado para selecionar uma das quatro opções, modo fácil (easy), modo médio (medium), modo difícil (hard) ou voltar para o menu inicial (back):



Fig.2 - Menu de Modos

2.2 Modo de Jogo

Neste modo o jogador é convidado a usar as teclas do teclado para mover o alien nas diferentes direções (A - esquerda, D - direita, S - baixo, W - cima) e o rato como mira dos projéteis e para os atirar para os aliens:

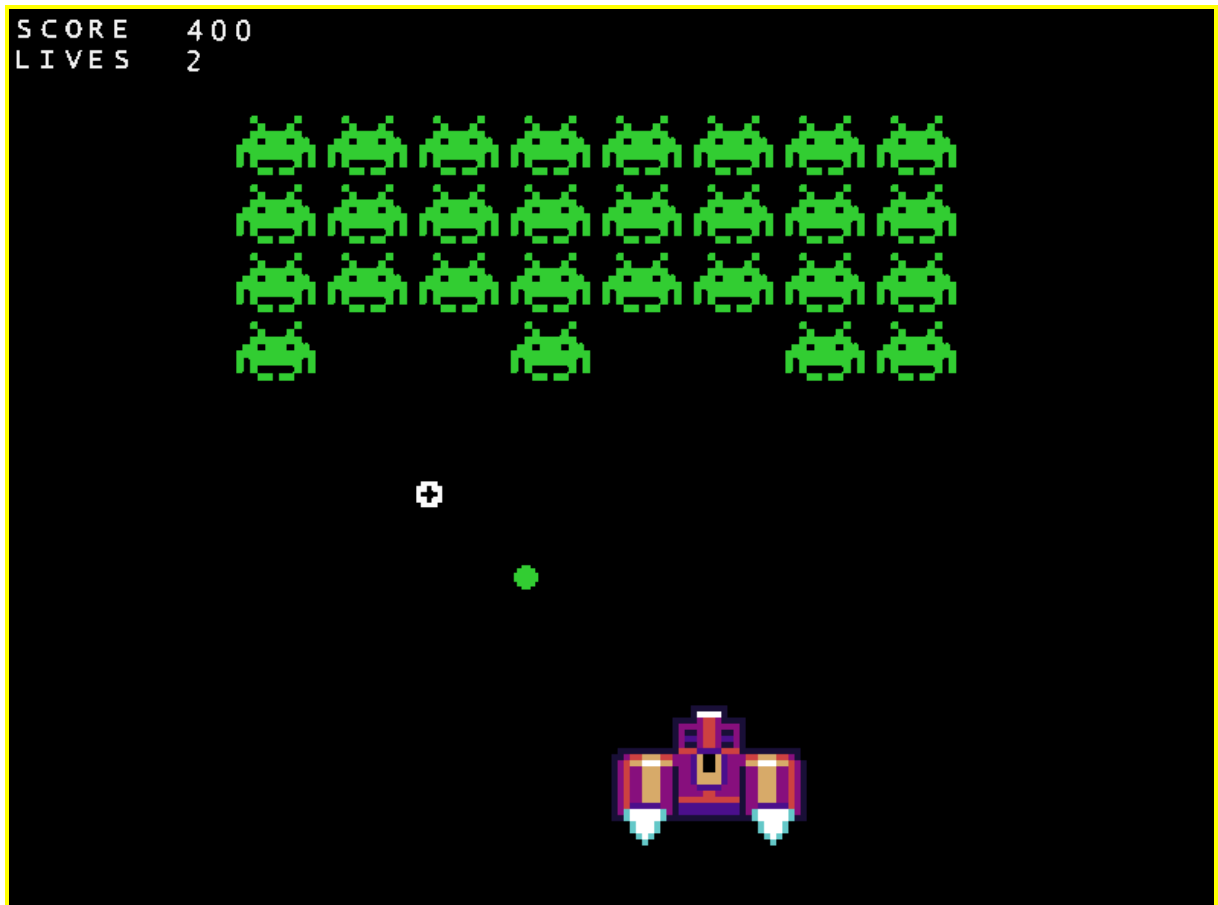


Fig.3 - Jogo

2.3 Menu Vitória/Derrota

Existem 2 formas de morrer no jogo: caso o jogador seja atingido várias vezes consecutivas pelos projéteis dos aliens, a ponto de ficar sem nenhuma das 3 vidas, ou caso não consiga matar os aliens a tempo suficiente, antes de eles atingirem a linha da nave. Quando ganha ou perde o jogador é redirecionado para um novo menu onde, caso ganhe, é apresentada a pontuação do último jogo e a sua melhor pontuação e onde é possível escolher com o rato ou com as setas do teclado se quer regressar ao menu principal ou se quer jogar novamente:

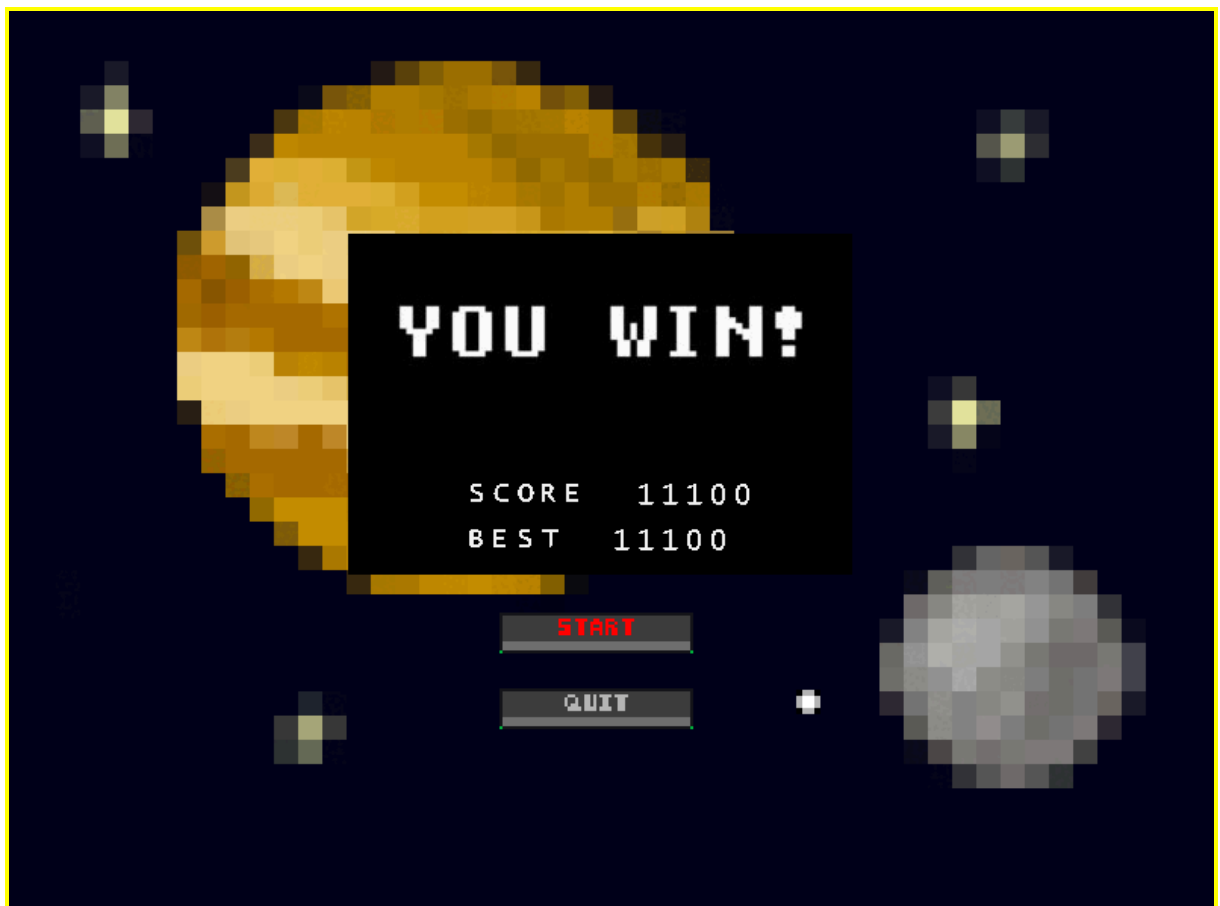


Fig.4 - Vitória

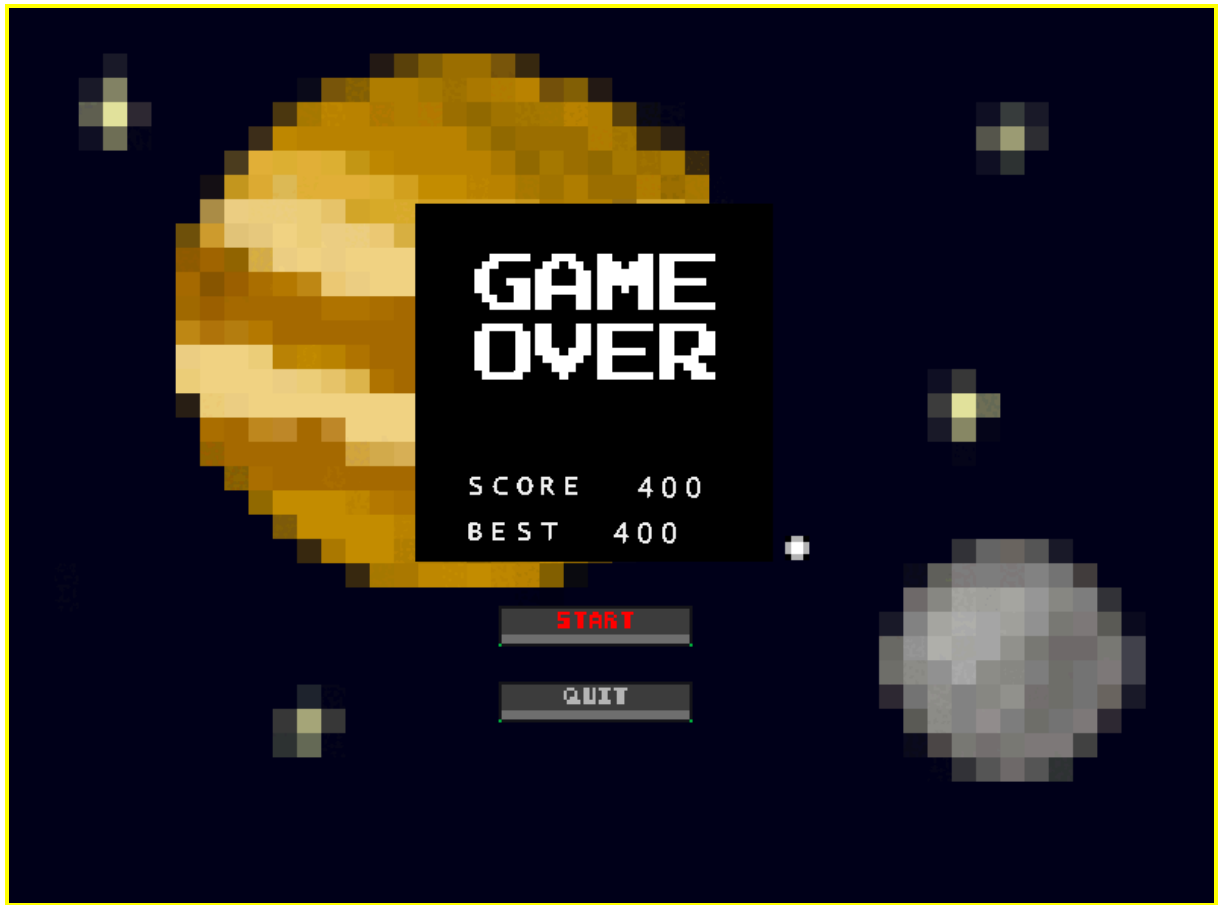


Fig.5 - Derrota

3. Estado do Projeto

3.1 Funcionalidades

Funcionalidades	Dispositivos	Implementado
Navegação nos menus	Keyboard, Mouse e Video Card	Sim
Deslocar a nave	Keyboard e Video Card	Sim
Mirar nos aliens	Mouse e Video Card	Sim
Atirar no aliens	Mouse e Video Card	Sim
Deslocar os aliens	Timer e Video Card	Sim
Mover Projéteis	Timer e Video Card	Sim
Mudar screen de acordo com as horas	RTC e Video Card	Sim

3.2 Dispositivos

Dispositivos	Funcionalidades	Interrupções
Timer	Atualizar placa vídeo, limitar tempo do jogo, desenhar aliens e colocá-los a disparar.	Sim
Keyboard (Controlo do jogo e input de texto)	Escrever o nome de jogador, navegar no menu e movimentar a nave espacial.	Sim
Mouse (botões e deslocamento)	Navegar nos menus, mirar nos aliens e disparar projéteis.	Sim
Video Card (double buffering, sprites com animação e deteção de colisão)	Desenhar com XPM's a nave e os aliens, desenhar os menus e animar sprites.	Não
RTC	Diferentes fundos para diferentes horas do dia	Não

3.2.1 Timer

O timer é utilizado para atualizar a placa de vídeo, temporizar o tempo do jogo para acabar ao fim de um certo tempo, desenhar os aliens de tempo em tempo e colocá-los a disparar a cada certos segundos dependendo da dificuldade.

A implementação das configurações, funcionalidades e interrupções do timer encontram-se no ficheiro timer.c.

3.2.2 Keyboard

O keyboard é utilizado para o utilizador poder escrever o seu nome de jogador, para selecionar opções nos menus através das teclas de setas e de ENTER e para movimentar a nave espacial, utilizando as teclas A e D para mover a nave para a esquerda e direita e W e S para cima e para baixo respetivamente.

A implementação das configurações, funcionalidades e interrupções do teclado encontram-se no ficheiro keyboards.c.

3.2.3 Mouse

Tal como o keyboard, o mouse permite ao utilizador navegar no menu, escolhendo entre as diferentes opções com o botão esquerdo. É utilizado também como mira nos aliens e para disparar os projéteis, utilizando o botão.

A implementação das configurações, funcionalidades e interrupções do mouse encontram-se no ficheiro mouse.c.

3.2.4 Video Card

A implementação das configurações, funcionalidades e interrupções do video card encontram-se no ficheiro video.c. Estamos a utilizar double buffering via copy.

Nós criamos através de xpm's a nossa fonte para utilizar na aplicação.

3.2.5 RTC

O RTC é utilizado para ler a hora do dia e apresentar diferentes sprites serem carregados para diferentes horas.

Para o rtc é enviado o comando "4" para ler a hora

A implementação das configurações, funcionalidades e interrupções do RTC encontram-se no ficheiro rtc.c.

4. Organização do Código / Estrutura

4.1 Timer Module - %

Este módulo contém as funções desenvolvidas no Lab2 das aulas práticas sobre o timer que puderam ser reutilizadas tendo em conta o propósito do nosso jogo. Estas funções permitem fazer a configuração e as interrupções do timer.

4.1.1 alien.c

O alien.c é o file em que nós desenhámos os xpm do alien e criámos uma forma de apagar o xpm do alien, criámos uma estrutura com a posição e estado (vivo ou morto) do alien, a função que criar e dispara os projéteis do alien, e funções relacionadas com a estrutura.

4.1.2 bg.c

O bg.c é o file em que nós desenhámos os xpm do background dos menus e criámos uma forma de apagar esses backgrounds.

4.1.3 buttons.c

O buttons.c é o file em que nós desenhámos os vários botões utilizados nos menus e criámos uma forma de apagar esses botões.

4.1.4 cursor.c

O cursor.c é o file em que nós desenhámos o cursor ou o alvo e criámos uma forma de apagar o cursor ou o alvo.

4.1.5 letters.c

O letters.c é o file em que nós desenhámos todas as letras e símbolos do keyboard e criámos uma forma de as apagar.

4.1.6 player.c

O player.c é o file em que nós desenhámos a spaceship e criámos uma forma de a apagar.

4.1.7 projectiles.c

O projectiles.c é o file em que nós desenhámos os projéteis e criámos uma forma de os apagar.

4.1.8 score.c

O score.c é o file em que nós desenhemos, atualizamos e guardamos as vidas e a pontuação do jogador.

4.1.9 sprites.c

O sprites.c é o file onde nós damos load de todos os sprites utilizados no jogo todos ao mesmo tempo.

5. Detalhes de Implementação

Código Dirigido por Eventos

O código faz uso extensivo de um modelo de programação dirigido por eventos. Isso é evidenciado pelo uso de interrupções para manipular eventos de teclado, mouse e timer. O manuseio de eventos é feito através de handlers específicos, como `mouse_ih`, `kbc_ih` e `timer_int_handler`, que são chamados quando ocorrem interrupções correspondentes. Este design assegura que o jogo responda rapidamente às ações do usuário e eventos de temporização.

Máquinas de Estado

O jogo é gerido através de uma máquina de estados, com estados como `MAINMENU`, `MODEMENU`, `PLAYING`, `GAMEOVER` e `WIN`. Cada estado possui um loop específico onde são processados diferentes tipos de entrada do usuário e ações específicas. Por exemplo, no estado `PLAYING`, o jogo atualiza a posição do jogador, movimento dos alienígenas, criação de projéteis e renderização da tela.

Geração de Frames

A função `double_buffer` é usada para implementar a técnica de double buffering, essencial para uma renderização suave e sem flickering. Esta técnica envolve desenhar os gráficos em um buffer fora da tela e, em seguida, copiar este buffer para a tela de uma só vez.

Detecção de Colisões

Um aspecto importante que não foi coberto nas aulas e que foi necessário aprender foi a detecção de colisões entre os projéteis do jogador e os alienígenas. Implementar a detecção de colisões envolveu calcular as posições dos projéteis e alienígenas e verificar se suas áreas se sobrepõem. Este processo é crítico para determinar acertos e contabilizar a pontuação do jogador.

6. Conclusões

Durante o desenvolvimento do nosso projeto, enfrentamos vários desafios que impactaram o nosso progresso, principalmente devido à carga de exames. Como resultado, não conseguimos implementar todas as funcionalidades planeadas, como uma tabela de pontuação (score board), e encontramos alguns problemas, como o carregamento lento do botão de voltar no menu.

Apesar dessas dificuldades, alcançamos a maioria dos nossos objetivos. Desenvolvemos com sucesso um jogo de Space Invaders com diferentes níveis de dificuldade, utilizando tanto o rato como o teclado para controlo. Esta conquista demonstra a nossa capacidade de aplicar conceitos de linguagens de baixo nível (C) e integrar periféricos (rato, timer, teclado, RTC) num contexto real.

No futuro, gostaríamos de adicionar a funcionalidade de tabela de pontuação para melhorar a experiência do jogador. Este projeto ensinou-nos lições valiosas sobre gestão de tempo, programação de baixo nível e uso de periféricos em situações práticas, competências que certamente serão úteis nos nossos próximos empreendimentos.