

# Faculdade de Engenharia da Universidade do Porto



## **SNAKE**

Projeto Final - LC - Turma 9 - Grupo 01

Realizado por:

Tomás Gomes, up202004393

Henrique Ferreira, up202007044

# Índice

Introdução.....	3
1. Instruções de Utilização.....	4
1.1 Menu Inicial.....	4
1.2 Board.....	4
1.2.1 Score.....	5
1.3 Scoreboard.....	5
1.4 Lost.....	6
2. Estado do Projeto .....	7
2.1 Timer.....	7
2.2 Teclado.....	8
2.3 Mouse.....	8
2.4 Graphics Card.....	8
2.5 RTC .....	9
3. Organização e Estrutura do Código .....	9
3.1 Timer Module (10%).....	9
3.2 Keyboard Module(10%) .....	9
3.3 Mouse Module(10%).....	9
3.4 Graphics Card Module(10%).....	9
3.5 RTC Module(5%) .....	9
3.6 Apple, Grass Module(5%) .....	9
3.7 Snake Module(15%) .....	10
3.8 Menu Module(15%).....	10
3.8 Game Module(20%).....	10
3.9 Function Call Graph.....	10
4. Detalhes de Implementação .....	10
5. Conclusões.....	11

## Introdução

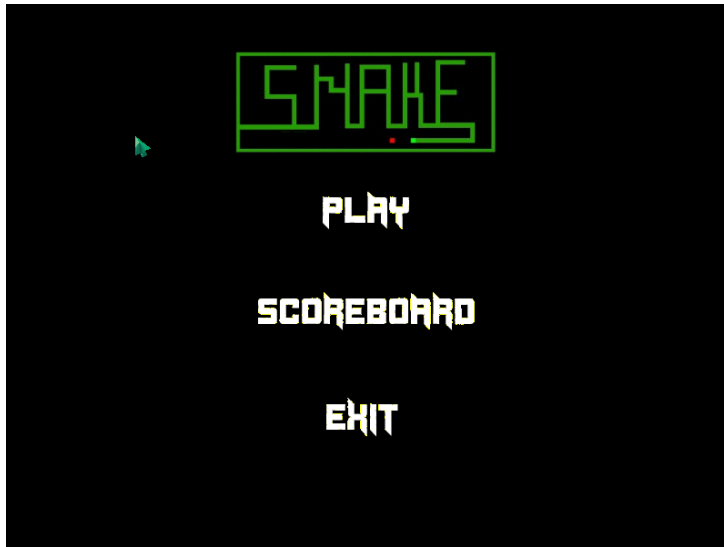
Para o nosso projeto final, decidimos criar um jogo em 2D, o clássico ‘Snake’.

O jogador é responsável por controlar uma cobra que, por sua vez, aumenta de tamanho cada vez que come uma maçã. A posição de uma maçã no quadro do jogo é gerada aleatoriamente, sendo que só pode existir uma única maçã no quadro do jogo. Assim, cada vez que a cobra come uma maçã, esta será gerada automaticamente noutra posição aleatória.

O objetivo do jogo é tentar comer o maior número de maçãs, de forma a atingir o maior tamanho possível. É de realçar que a cobra não pode embater em si própria nem sair fora da área de jogo. Caso aconteça, o jogador perde e terá de reiniciar o jogo, voltando a cobra ao seu tamanho inicial.

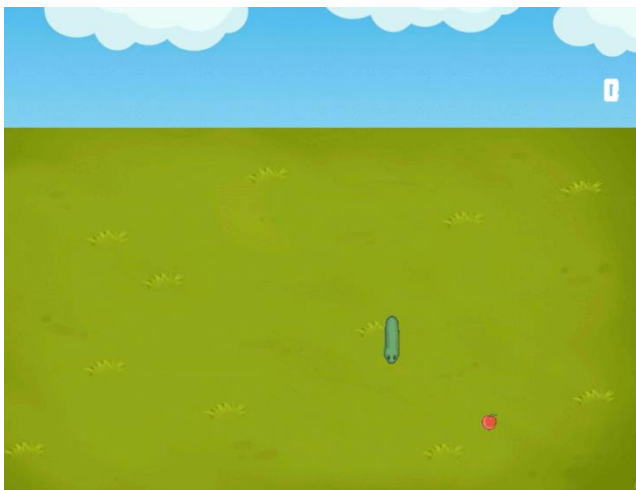
# 1. Instruções de Utilização

## 1.1 Menu Inicial



Ao iniciar o jogo, é mostrado o menu inicial, onde é possível escolher para jogar (PLAY), ver as pontuações de jogadores anteriores (SCOREBOARD) ou sair do jogo (EXIT). Para navegar neste menu deve ser utilizado o rato, movendo o cursor e clicando no botão esquerdo para seleccionar a opção desejada.

## 1.2 Board



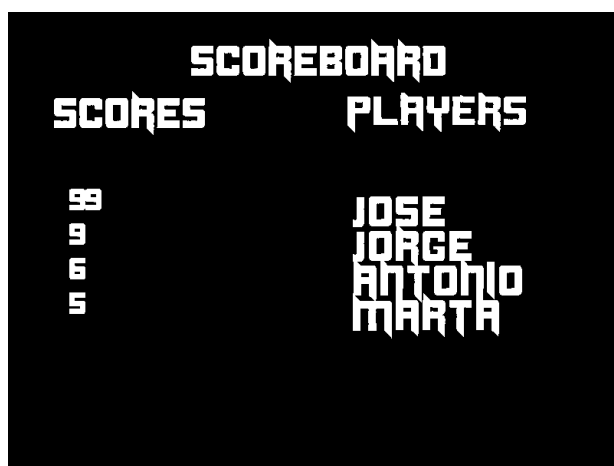
Quando se inicia o jogo, a cobra começa por se mexer horizontalmente para a direita automaticamente e uma maçã é gerada. Para mudar o movimento atual da cobra, deve-se usar o teclado premindo as teclas WASD (W para movimento vertical para cima, S para movimento vertical para baixo, A para movimento horizontal para a esquerda e D para movimento horizontal para a direita). Enquanto uma tecla não for premida, a cobra manter-se-á no seu tipo de movimento. O board do jogo é constituído por um fundo (ao que chamamos de grass), um céu, uma cobra e uma maçã.

### 1.2.1 Score



Existe um contador no topo do ecrã, o SCORE, que a cada vez que a cobra come uma maçã, é incrementado. Assim, guarda-se a informação sobre a pontuação do jogador para no final ser possível adicionar ao Scoreboard. O céu mudará consoante a hora atual.

### 1.3 Scoreboard



Esta janela tem o propósito de apresentar ao jogador todas as pontuações já feitas previamente. Para voltar ao menu inicial basta premir a tecla ESC.

## 1.4 Lost



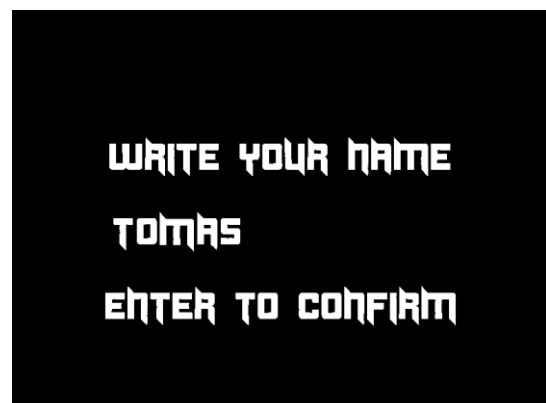
É possível perder o jogo de duas formas:

- Quando a cobra ultrapassa o board do jogo, ou seja, quando a sua posição X é maior que 800 ou a sua posição Y é maior que 600 ( uma vez que se usa o modo gráfico 0x115 )
- Quando a cobra embate em si própria, ou seja, quando a posição X e Y da cabeça coincide com a posição X e Y do corpo/cauda.

É de salientar que não é possível ganhar o jogo. O objetivo é tentar ficar no topo do Scoreboard, fazendo o maior número possível de pontos.

Finalmente, é apresentada a pontuação do jogador e, após o utilizador premir a tecla ESC, será apresentada uma tela onde se pede para o utilizador escrever o seu nome (para posteriormente ser adicionado ao Scoreboard).

A imagem da esquerda corresponde à tela após ser premido a tecla ESC, a imagem da direita é após o utilizador escrever o seu nome. É de realçar que a cada tecla que o utilizador prime, esta será imprimida no ecrã em forma de XPM, para além de que é possível apagar as letras utilizando a tecla BACKSPACE.



## 2. Estado do Projeto

Utilizou-se os dispositivos Timer, Keyboard, Mouse, Graphics Card e Real Time Clock, sendo o Serial Port o único dispositivo não implementado. Em todos os dispositivos foi usado o modo INTERRUPTIONS.

Funcionalidade de cada dispositivo:

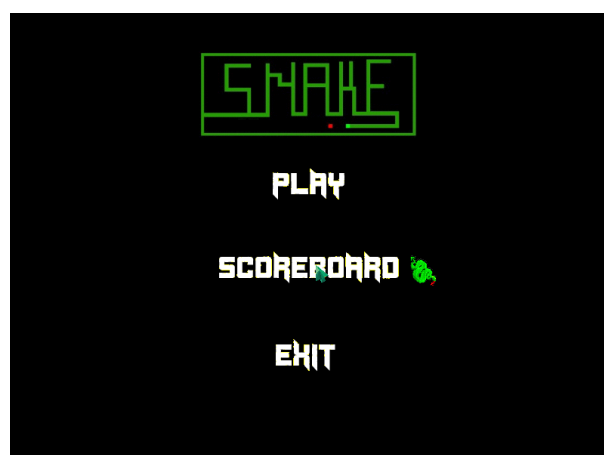
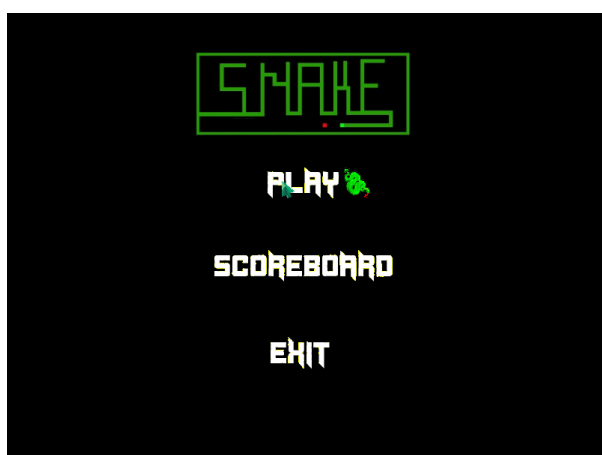
- Timer – Controlo da frame-rate. Eventos do jogo e animações.
- Teclado – Movimentação da cobra. Navegação nos menus.
- Rato – Navegação nos menus.
- Graphics Card – Interface do jogo, menu, etc.
- Real Time Clock – Guardar data e hora do final de um jogo.

### 2.1 Timer

A implementação do timer encontra-se no **timer.c**.

O timer é utilizado para controlar o frame-rate do ecrã, fazendo atualizações no ecrã 15 vezes por segundo.

No menu (função [MenuInterruptHandler](#)), o timer é utilizado para verificar se o cursor do rato está sobre algum dos botões, adicionando uma cobra ao lado do direito do botão caso o cursor esteja sobre o botão em questão.



Também é utilizado para eliminar a imagem do cursor do rato e voltar a desenhá-la (já com a sua posição atualizada).

No modo de jogo (função [PlayInterruptHandler](#)), o timer é utilizado para gerir todos os

eventos. Permite fazer a animação da cobra, bem como desenhar todos os elementos (XPM's), como a maçã, a pontuação, o fundo, etc. Neste modo, também é usado para verificar colisões da cobra com a maçã ou com as bordas do ecrã.

## 2.2 Teclado

A implementação do teclado encontra-se no **keyboard.c**.

O teclado é usado para movimentar a cobra usando as teclas WASD. Através do make-code e break-code de cada tecla, foi possível definir em que direção a cobra se deslocaria.

Na fase de jogo LOST a tecla ESC permite avançar para a próxima fase ( SAVE ) onde será pedido o nome do utilizador. Quando é apresentado o scoreboard, também é possível voltar para o menu inicial premindo a tecla ESC.

## 2.3 Mouse

A implementação do rato encontra-se no **mouse.c**.

O rato é utilizado para navegar no menu usando um cursor. Através da função [mouse\\_events](#) é obtido o evento do rato, permitindo saber se foi premido algum botão.

## 2.4 Graphics Card

A implementação da placa de vídeo encontra-se no **video\_gr.c**.

Para o projeto foi utilizado o modo de vídeo 0x115.

Foi utilizada a técnica de double buffering, onde nas interrupções do timer se chama a função [double\\_buffer](#) para copiar o conteúdo da memória secundária para a memória principal.

Foram usados XPMs para todas as imagens no jogo:

- Para a cobra, definiu-se um XPMs para a cauda , corpo e cabeça. Para cada parte da cobra, há um XPM para cada tipo de movimento (direita, esquerda, cima, baixo). Para além disso, também se definiu um XPM para quando a cobra muda o seu movimento de vertical para horizontal e vice-versa.
- Para o cursor do rato, maçã e fundo de jogo também se definiu um XPM.
- Para o menu, o logotipo é representado por um XPM e o resto do menu é todo ele um XPM.
- Para a pontuação, fez-se um XPM para os números de 0-9.

Para a transparência dos XPMs é usada a cor 'None'.



## 2.5 RTC

A implementação do real time clock encontra-se no **rtc.c**.

O RTC é usado para saber a data e hora, mudando o céu do board consoante a hora do dia em que se encontra.

São utilizadas interrupções para atualizar o tempo a cada segundo, onde a leitura dos registos é feita em modo binário.

## 3. Organização e Estrutura do Código

### 3.1 Timer Module (10%)

Funções desenvolvidas no lab2. Ambos contribuíram para a realização deste módulo.

### 3.2 Keyboard Module(5%)

Funções desenvolvidas no lab3. Ambos contribuíram para a realização deste módulo.

### 3.3 Mouse Module(10%)

Funções desenvolvidas no lab4. Também foi adicionada uma struct 'Cursor' para ajudar a guardar informação sobre o cursor do rato, bem como funções complementares que verificam os eventos do rato, colisões com botões do menu, etc. Ambos contribuíram para a realização deste módulo.

### 3.4 Graphics Card Module(10%)

Funções desenvolvidas no lab5. É de salientar que se implementou a função [vbe\\_get\\_info](#) e [double\\_buffer](#). Ambos contribuíram para a realização deste módulo.

### 3.5 RTC Module(5%)

Implementação do RTC. Este módulo foi desenvolvido pelo aluno Tomás Gomes.

### 3.6 Apple, Grass Module(5%)

Módulo relativo à maçã e fundo do jogo (board). Ambos contribuíram para a realização deste módulo.

### 3.7 Score Module (5%)

Módulo relativo ao desenho da pontuação no ecrã. Ambos contribuíram para a realização deste módulo.

### 3.8 Snake Module(15%)

Módulo relativo à cobra. É neste módulo que são verificadas as colisões da cobra com a maçã, bem como onde se define as animações das sprites da cobra. Este módulo foi desenvolvido pelo aluno Henrique Ferreira.

### 3.8 Menu Module(15%)

Módulo relativo aos menus do jogo (menu principal e menu de 'Game Lost'). Aqui encontram-se as funções que dão load aos menus. Este módulo foi desenvolvido pelo aluno Tomás Gomes.

### 3.9 Game Module(20%)

Este é o módulo mais importante. É nele onde se encontra o ciclo de interrupções que permite a execução do programa. O jogo está dividido em estados {MAINMENU, LOST, SCOREBOARD, EXIT} e o *handling* das interrupções é feito dependendo do estado de jogo atual.

### 3.10 Function Call Graph

Está no html do doxygen.

## 4 Detalhes de Implementação

Tentou-se seguir uma espécie de *implementação object-oriented*, com o intuito de facilitar a leitura do código. Assim, cada unidade do jogo ( por exemplo, cobra, maçã, etc.) tem a sua própria implementação do seu desenho, load, delete, bem como coordenadas no board, a sua `xpm_image_t`, etc.

As colisões de rato com botões e cobra com maçã são detetadas pelas coordenadas de cada objeto. Caso o X e o Y coincidam, então há uma colisão.

Para representar o modelo de controlo de flow de execução do jogo usou-se uma **state machine** para saber em que estado o jogo está.

Assim, é possível diferenciar qual é o handling das interrupções que deve ser feito para cada estado de jogo.

Quanto à parte gráfica, a função `load_xpm()` para cada XPM só é chamada uma vez, guardando posteriormente a `xpm_image_t` retornada no argumento da função, de forma a garantir maior eficiência e performance do jogo. O XPM do menu e os números para a pontuação foram implementados com recurso ao site <https://fixthephoto.com/pt/gimp-online.html>. Tanto o logotipo, o cursor, a cobra, a maçã foram retirados de sites de uso grátis. Maioritariamente utilizou-se um fundo preto, ao qual se foram adicionando/removendo XPMs, prevalecendo no ecrã o elemento que for desenhado mais recentemente.

Para o RTC, utilizou-se dois arrays para guardar informação tanto para as horas como para a data atual.

## 5. Conclusões

A elaboração deste projeto revelou-se difícil, tendo em conta que foi feita apenas por 2 pessoas. No entanto, acabou por nos ensinar bastante, uma vez que a forma de programar difere daquilo que estávamos habituados. Para nós, um aspeto negativo é o facto de não existir labs para RTC. Isso dificultou a implementação desses dispositivos no projeto.