

**U. PORTO**

**FEUP FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO**



## **Sudoku Mania**

### **Relatório de projeto**

Laboratório de Computadores

Nuno Castro up202003324  
Angela Cruz up201806781

## Índice

Índice de figuras	2
Índice de tabelas	2
1 – Instruções de utilização	3
1.1 Ecrã / Menu inicial	3
1.2 Seleção de dificuldade (Modo Time-Attack)	5
1.3 Ecrã de jogo	6
2 – Funcionalidades Implementadas	7
2.1 Timer	8
2.2 Keyboard	8
2.3 Mouse	8
2.4 Placa gráfica	8
3 – Estruturação do código	9
3.1 libs	10
3.1.1 kbc	10
3.1.2 timer	10
3.1.3 utils	10
3.1.4 vbe	10
3.2 game	10
3.3 xpm	10
4 – Detalhes de implementação	12
4.1 Interrupts	12
4.2 Events	12
4.3 XPM	12
4.4 Multiple Buffering	12

## Índice de figuras

Figura 1 Ecrã inicial.....	3
Figura 2 Exemplo de botão destacado.....	4
Figura 3 Menu de dificuldades .....	5
Figura 4 Exemplo de botão destacado.....	6
Figura 5 Tela de jogo .....	6
Figura 6 Exemplo de célula destacada .....	7
Figura 7 Imagens referentes a vitória e derrota.....	7
Figura 8 Gráfico de chamada de funções.....	9

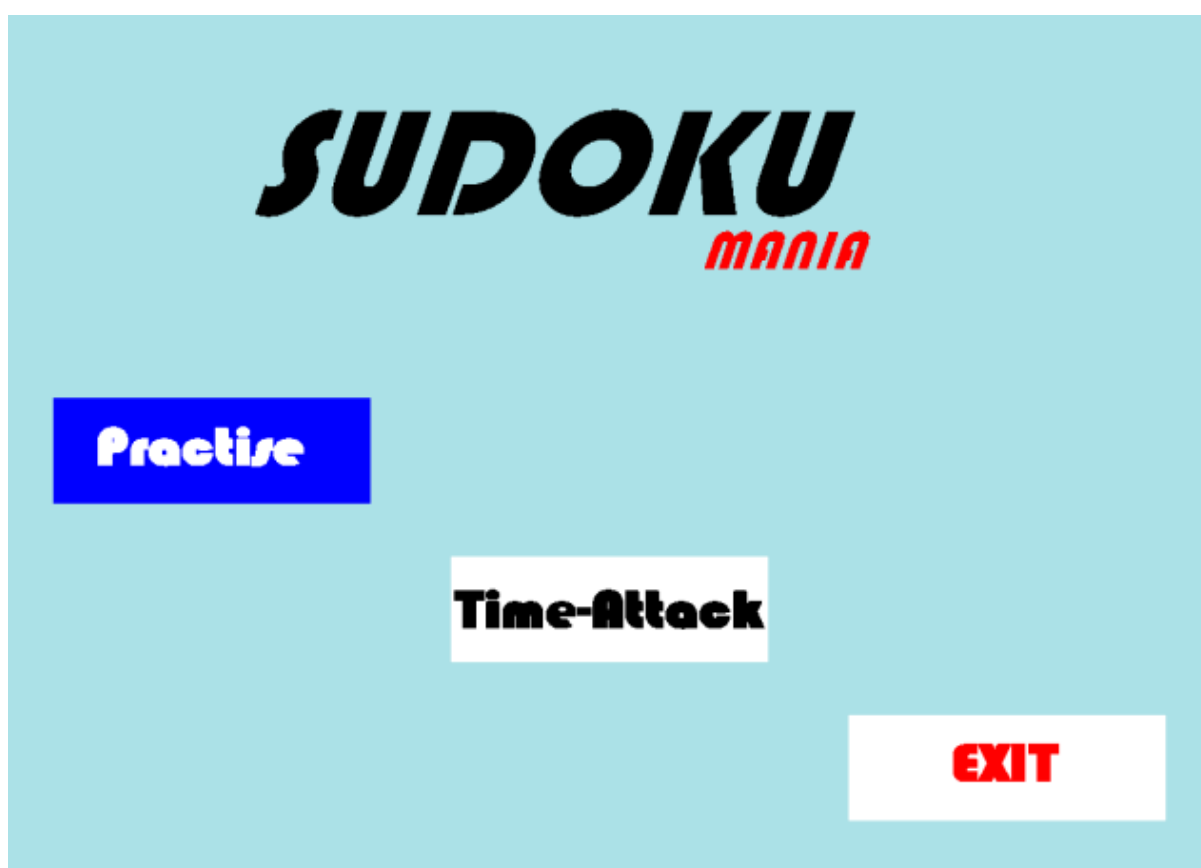
## Índice de tabelas

Tabela 1 Tabela de periféricos utilizados .....	7
Tabela 2 Tabela de pesos.....	11

# 1 – Instruções de utilização

## 1.1 Ecrã / Menu inicial

Ao iniciar o jogo é apresentado o menu inicial ao utilizador, menu este que lhe permite seleccionar entre os dois modos de jogo ou sair do programa.



*Figura 1 Ecrã inicial*

Para navegar neste menu, os seguintes controlos são:

- as setas direccionais do teclado ou o rato para movimento;
- a tecla enter ou o botão esquerdo do rato para seleccionar;
- a tecla Esc que funciona como atalho para seleccionar a opção EXIT.

Ainda relativo ao menu inicial, existem 3 botões que poderão ser seleccionados:

<b>Practise</b>	Procede para o modo de jogo Practise;
<b>Time-Attack</b>	Procede o modo de jogo Time-Attack;
<b>EXIT</b>	Sai do jogo.

Quando um botão se apresenta com o fundo azul em vez de o normal branco significa que este é o atual botão pronto para seleção.



*Figura 2 Exemplo de botão destacado*

#### 1.1.1 Modo Practise

Este modo do jogo funciona como um modo de jogo livre em que é apresentado ao utilizador um sudoku para resolver e o jogo só termina quando este for completado ou o utilizador desistir.

Enquanto isso, um cronómetro vai contando o tempo necessário pelo utilizador para a resolução do puzzle.

Selecionando este modo, o programa procede imediatamente para a tela de jogo.

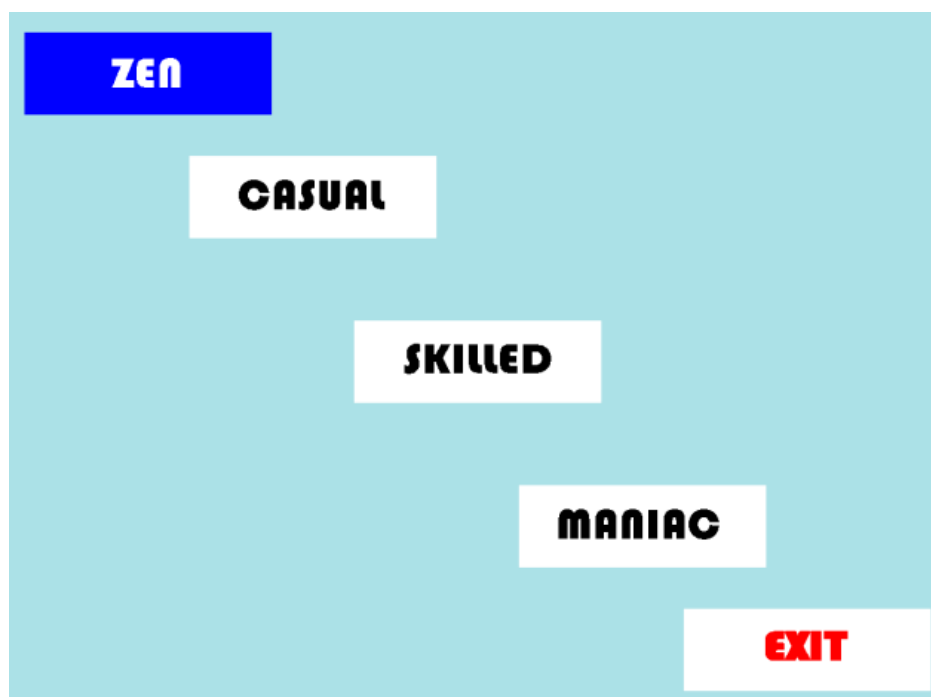
#### 1.1.1 Modo Time-Attack

Neste modo de jogo, modo de desafio, é igualmente apresentado um sudoku por resolver ao utilizador, porém, ao contrário do modo anterior, existe um temporizador e se o tempo estipulado pela dificuldade selecionada termina sem o jogador resolver o puzzle, este perde e termina o jogo. Caso o jogador resolva ou desista o jogo termina também.

Selecionando este modo, o programa procede imediatamente para um menu de seleção de dificuldade.

## 1.2 Seleção de dificuldade (Modo Time-Attack)

O seguinte menu é apresentado ao utilizador caso este tenha escolhido o modo de jogo Time-Attack no ecrã inicial. É aqui que é feita a seleção da dificuldade de jogo.



*Figura 3 Menu de dificuldades*

Os controlos de navegação neste menu são idênticos aos de navegação no menu inicial, mencionados em 1.1.

Podem ser seleccionadas as seguintes opções, que estão apresentadas por ordem crescente de dificuldade (com excepção de EXIT), traduzido no tempo disponível para completar o desafio:

<b>ZEN</b>	Dificuldade Zen (fácil), 10 minutos;
<b>CASUAL</b>	Dificuldade Casual (intermédio), 7,5 minutos;
<b>SKILLED</b>	Dificuldade Skilled (difícil), 5 minutos;
<b>MANIAC</b>	Dificuldade Maniac (muito difícil), 2,5 minutos;
<b>EXIT</b>	Voltar ao menu anterior.

Igualmente ao menu inicial, quando o botão apresenta fundo azul é o botão pronto a ser selecionado.



Figura 4 Exemplo de botão destacado

### 1.3 Ecrã de jogo

Na imagem que se segue pode-se observar a tela de jogo. Esta é idêntica para ambos os modos de jogo com a diferença que no modo *Practise* o temporizador vem carregado com o valor 00:00 enquanto que no *Time-Attack* o tempo referente à dificuldade escolhida.

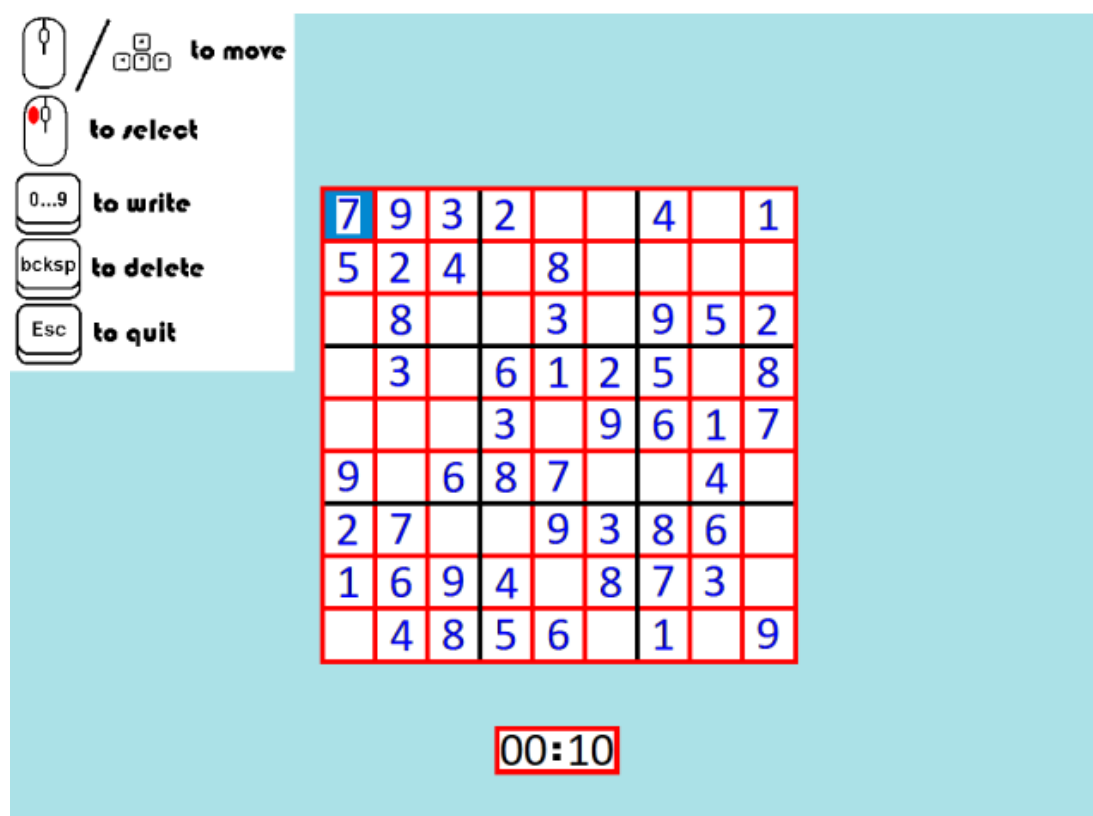


Figura 5 Tela de jogo

Os controlos de jogo, que também se apresentam na tela de jogo são:

- as setas direcionais do teclado ou o rato para movimento;
- o botão esquerdo do rato para selecionar célula;
- os botões numéricos do teclado para escrita da célula;

- o botão Backspace para limpar célula;
- a tecla Esc para desistir do jogo.

A célula representativa da posição do jogador na grelha é indicada pelo fundo azul ao invés do normal preenchimento branco.



*Figura 6 Exemplo de célula destacada*

Quando um jogador completa ou perde um desafio são apresentadas as seguintes imagens, respetivamente.



*Figura 7 Imagens referentes a vitória e derrota*

## 2 – Funcionalidades Implementadas

*Tabela 1 Tabela de periféricos utilizados*

Device	Objetivo	Método
Timer	Cronómetro/temporizador	Ints
KBD	Movimento, escrita, seleção	Ints
Mouse	Movimento, seleção	Ints
Video card	Geração de imagens	-



## 2.1 Timer

O timer é utilizado com o propósito de funcionar como cronómetro no modo Practise, ou como temporizador no modo Time-Attack.

Para a subscrição dos interrupts usou-se a função `timer_subscribe_int()`, para remover a subscrição `timer_unsubscribe_int()` e para gerir os interrupts usou-se a função `timer_int_handler()`.

## 2.2 Keyboard

O keyboard, utilizado para movimento nos menus e jogo, escrita na grelha e seleção de opções foi configurado para utilizar interrupts. Para a subscrição dos interrupts usou-se o `kbc_subscribe()`, para remover a subscrição `kbc_unsubscribe_int()` e para gerir os interrupts usou-se o `kbc_ih()`.

## 2.3 Mouse

O mouse foi configurado para movimento e seleção de botões nos menus e jogo. Recorremos a interrupts para este efeito. Para implementar estas funcionalidades criamos as funções `mouse_packet_builder()`, `mouse_ih()`, `mouse_subscribe_int()`, `kbc_unsubscribe_int()` e o `get_mouse_pos()` para descobrir a posição do mouse num instante e comparar se poderá estar dentro de uma célula ou botão.

## 2.4 Placa gráfica

A placa gráfica foi configurada para trabalhar com a resolução 1152x864, direct encoding e multiple buffering. Usamos 3 buffers para desenvolver os menus e o layout do sudoku recorrendo a várias funções, mas especialmente a `vg_set_pixel()`, para o print dos elementos gráficos no modo de vídeo de maneira user friendly.

### 3 – Estruturação do código

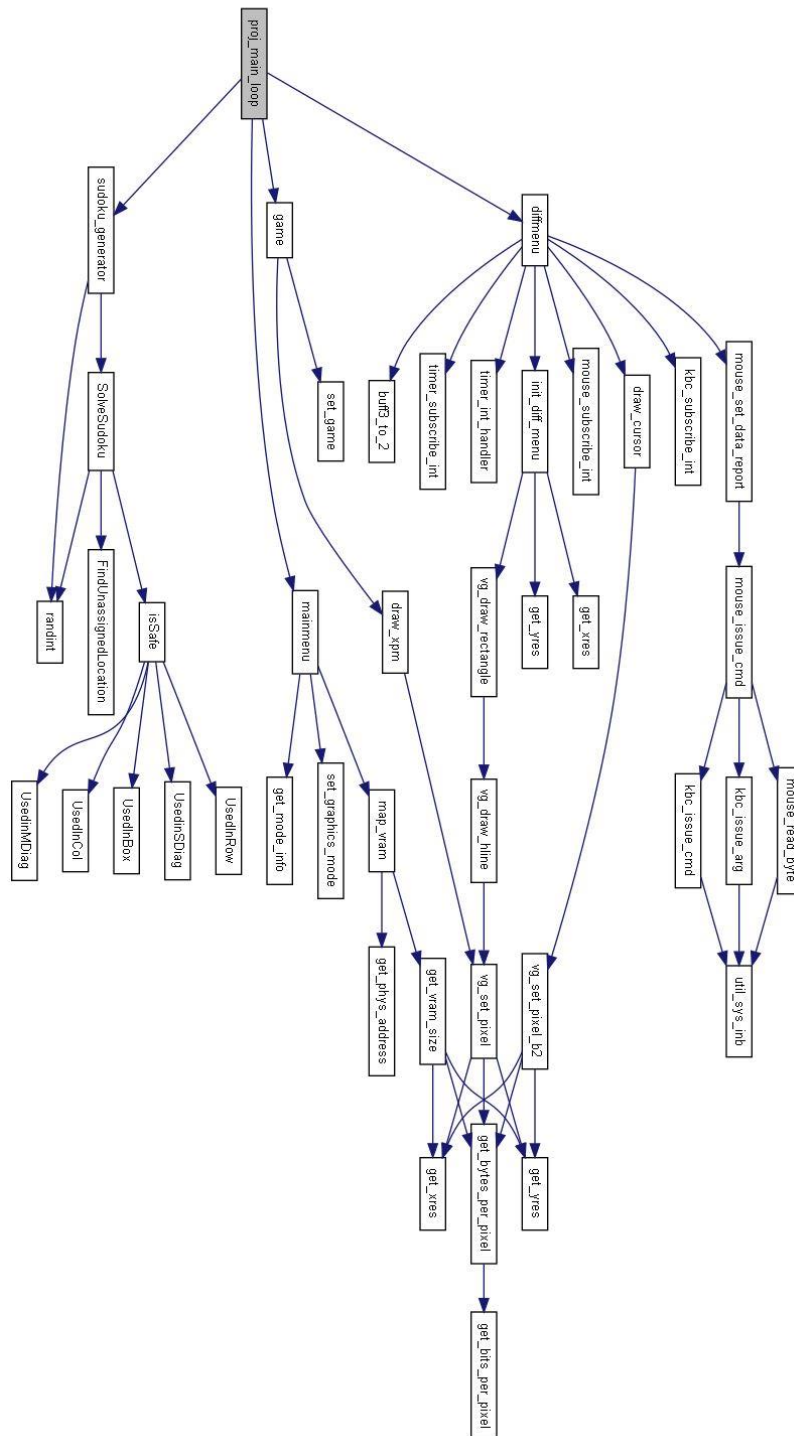


Figura 8 Gráfico de chamada de funções

## 3.1 libs

Neste módulo encontram-se funções importantes ao funcionamento do programa mas que não lhe são inerentes, isto é, funções gerais não específicas para este programa.

### 3.1.1 kbc

Funções de tratamento do rato e teclado.

### 3.1.2 timer

Funções de tratamento do timer.

### 3.1.3 utils

Funções para auxílio ao tratamento de scancodes.

### 3.1.4 vbe

Funções para tratamento do ambiente gráfico.

## 3.2 game

Módulo constituído por funções específicas ao programa para geração de puzzles, tratamento de menus e do jogo em si.

## 3.3 xpm

Módulo onde se encontram armazenados os elementos gráficos utilizados pelo programa.

*Tabela 2 Tabela de pesos*

	Peso (%)	Contribuição Angela	Contribuição Nuno
Kbc	5	Desenvolvido nos labs e ligeiras adaptações	
Mouse	5		
Event_handler	5		
Timer	5		
Utils	2		
Vbe_functions	10		
Sudoku	35	Funções gerador de sudokus e mouse ints	Mainmenu(), diffmenu(), Game(), exceto mouse ints
Grid	25	Funções get_mouse	Resto
proj	3	1/2	
xpm	5	Algumas contribuições	Resto

## 4 – Detalhes de implementação

### 4.1 Interrupts

Optou-se por utilizar interrupções visto que, habitualmente, os periféricos são bastantes mais lentos que o processador e através de interrupções, o periférico notifica o processador quando completa a sua operação o que nos pareceu ser pertinente para uma mais eficiente execução do programa.

### 4.2 Events

Devido à utilização de vários periféricos e operações distintas, o recorrer a eventos para a programação do jogo revelou-se ser um fator importante ao seu funcionamento, fazendo com que o código seja executado conforme estes eventos, já que a maior parte das vezes são-lhe assíncronos.

### 4.3 XPM

No que toca a elementos gráfico optou-se por utilizar xpm porque permite armazenar esse mesmo elemento num ficheiro de texto.

### 4.4 Multiple Buffering

Um dos entraves na programação deste jogo foi a performance gráfica. Para superar este problema optamos por recorrer a multiple buffering em que utilizamos mais que um buffer e enquanto uma imagem está a ser mostrada a seguinte já está a ser produzida num outro. Esta técnica permitiu-nos ganhos na fluidez da imagem e na performance do jogo também.