# Índice

[Índice 1](#_Toc439457857)

[Instruções de utilização 2](#_Toc439457858)

[Estado do projecto 3](#_Toc439457859)

[Timer 4](#_Toc439457860)

[Teclado 5](#_Toc439457861)

[Rato 6](#_Toc439457862)

[Placa gráfica 7](#_Toc439457863)

[RTC 8](#_Toc439457864)

[UART 9](#_Toc439457865)

[Organização/estrutura do código 10](#_Toc439457866)

[Game 11](#_Toc439457867)

[Graphics 12](#_Toc439457868)

[Graphics\_vbe 13](#_Toc439457869)

[Graphics\_video\_gr 14](#_Toc439457870)

[KBD 15](#_Toc439457871)

[Logger 16](#_Toc439457872)

[Main 17](#_Toc439457873)

[Menu 18](#_Toc439457874)

[Mouse 19](#_Toc439457875)

[Player 20](#_Toc439457876)

[Read\_xpm 21](#_Toc439457877)

[RTC 22](#_Toc439457878)

[Sprite 23](#_Toc439457879)

[Timer 24](#_Toc439457880)

[UART 25](#_Toc439457881)

[Gráfico de chamada de funções 26](#_Toc439457882)

[Detalhes de implementação 27](#_Toc439457883)

[Conclusões 28](#_Toc439457884)

# Instruções de utilização

# Estado do projecto

|  |  |  |  |
| --- | --- | --- | --- |
| Dispositivo | Para quê: | Interrupções | Varrimento |
| Timer | Suspender o jogo por algum tempo | Y | N |
|  |  |  |  |
|  |  |  |  |
| Teclado | Ler as escolhas do utilizador | Y | Y |
|  | Movimentar o copo durante o jogo |  |  |
|  |  |  |  |
| Rato | Movimentar o copo durante o jogo | Y | N |
|  |  |  |  |
|  |  |  |  |
| Placa gráfica | Mostrar o ambiente de jogo | N | Y |
|  |  |  |  |
|  |  |  |  |
| RTC | Contar o tempo de jogo e da jogada | N | Y |
|  |  |  |  |
|  |  |  |  |
| UART |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Timer

Relativamente ao “timer”, este está a ser utilizado em modo de interrupção e serve para contar um certo tempo de paragem do jogo.

As funções que fazem uso das potencialidades do “timer” estão declaradas em “Timer.h” e definidas em “Timer.c”

(indicar em que ficheiros e linhas são chamadas funções do “timer”)

## Teclado

Relativamente ao teclado, este está a ser utilizado tanto em modo de interrupção como em modo de varrimento e serve para ler as teclas premidas pelo utilizador e com base nas teclas pressionadas movimentar as opções de um menu, movimentar os dados e escolher a opção de abortar ou continuar o jogo no final de cada jogada.

As funções que fazem uso das potencialidades do teclado estão declaradas em “Keyboard.h” e definidas em “Keyboard.c”

(indicar em que ficheiros e linhas são chamadas funções do teclado)

## Rato

Relativamente ao rato, este está a ser utilizado em modo de interrupção e serve para ajustar a posição de um cursor no ecrã ou de uma forma mais correcta na memória gráfica; faz uso tanto do sensor do deslocamento como dos botões do rato sendo que os botões servem para selecionar um objecto e a posição serve para deslocar o cursor ao longo do ecrã.

As funções que fazem uso das potencialidades do rato estão declaradas em “Mouse.h” e definidas em “Mouse.c”

(indicar em que ficheiros e linhas são chamadas funções do rato)

## Placa gráfica

O modo de vídeo utilizado na placa gráfica é o modo 0x105 cuja resolução é 1024x768. Este modo suporta 256 cores das quais só 64 são utilizáveis, porque as cores 64 a 255 têm uma tonalidade muito próxima do preto. Cada cor é codificada em x bits. Está a ser utilizado double buffering. Há objectos que se movem (sprites animados), mas não é feita detecção de colisões fontes Não foram utilizadas funções para alterar a palette.

As funções que fazem uso das potencialidades da placa gráfica estão declaradas em “Graphics.h”, “Graphics\_vbe.h” e “Graphics\_video\_gr.h” e definidas em “Graphics.c”, Graphics\_vbe.c” e “Graphics\_video\_gr.c”

(indicar em que ficheiros e linhas são chamadas funções da placa gráfica)

## RTC

Relativamente ao real-time clock, este está a ser utilizado em modo de varrimento e serve para contar o tempo de uma jogada e contar o tempo total do jogo, sendo que apenas é lida a data e a hora.

As funções que fazem uso das potencialidades do RTC estão declaradas em “RTC.h” e definidas em “RTC.c”

(indicar em que ficheiros e linhas são chamadas funções do RTC)

## UART

Relativamente à porta série, esta está a ser utilizada em modo de varrimento e serve para transmitir as pontuações e o tempo de uma jogada. Estão a ser utilizados como parâmetros de comunicação 8 bits por caractere, 1 stop bit e 9600 bits por segundo. A informação é transmitida no final de cada jogada.

As funções que fazem uso das potencialidades da porta série estão declaradas em “UART.h” e definidas em “UART.c”

(indicar em que ficheiros e linhas são chamadas funções da UART)

# Organização/estrutura do código

## Game

O ficheiro “Game.c” possui código para “atirar os dados” e fazer uma jogada, gerindo o seu ambiente gráfico.

O responsável por este código é Carolina Centeio.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 5%.

## Graphics

O ficheiro “Graphics.c” possui código para “desenhar um quadrado”, desenhar um XPM, preencher o fundo, iniciar o modo gráfico, transferir a memória de trabalho para a VRAM, desenhar os diversos menus iniciais, ler e imprimir strings, desenhar pontuações menores que 100, desenhar o ambiente inicial de jogo, atualizar as posições dos dados, desenhar os dados, repor o ambiente inicial de jogo, desenhar o menu de opção de continuação ou finalização do jogo.

O responsável por este código é Carolina Centeio.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 20%.

## Graphics\_vbe

O ficheiro “Graphics\_vbe.c” possui o código que permite obter as características da placa de vídeo.

O responsável por este código é ?.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 5%.

## Graphics\_video\_gr

Este ficheiro possui duas estruturas de dados: a estrutura vbe\_mode\_info\_t (que serve para receber e guardar informação sobre um dado modo da placa gráfica) e a estrutura VBEInfoBlock (que serve para receber e guardar as propriedades da placa gráfica).

O ficheiro “Graphics\_video\_gr.c” possui o código que permite iniciar o modo de vídeo, inicializar algumas variáveis relativas à placa gráfica e desligar o modo de vídeo.

O responsável por este código é ?.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 10%.

## KBD

O ficheiro “KBD.c” possui o código que permite subscrever e des-subscrever as interrupções do KBC, ler por varrimento um scancode (que estará completo se só for composto por 8 bits senão ainda falta a outra metade) e ler um scancode completo por interrupção.

O responsável por este código é Luís Barbosa.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 15%.

## Logger

(se for para colocar)

O responsável por este código é ?.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de ?%.

## Main

Serve para iniciar o programa, permitir operações de I/O ao programador, iniciar a semente da função de geração de números aleatórios, chamar as funções que iniciam o modo gráfico, chamar a função que inicia o ambiente de jogo e terminar o modo gráfico.

O responsável por este código é ?.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 3%.

## Menu

O ficheiro “Menu.c” contém uma função que mostra o menu inicial de forma a ser possível iniciar o jogo ou terminar o programa e uma função que é chamada no final de cada jogada perguntando se deseja continuar ou sair do jogo.

O responsável por este código é ?.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 5%.

## Mouse

O ficheiro “Mouse.c” contém as funções necessárias para interagir com o rato, mais concretamente o KBC e receber a informação produzida pelo rato.

O responsável por este código é Luís Barbosa.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 8%.

## Player

“Player” é uma estrutura de dados que guarda informação sobre um jogador, nomeadamente “Saldo”, “Nome”, “Pontuação”, “Activo” (se o jogador está neste momento a jogar ou não) e “Tempo de jogo” (do jogador).

O ficheiro “Player.c” contém as funções necessárias para criar um jogador, coloca-lo inactivo, adicionar pontuações, verificar se é possível fazer uma aposta e apagar o jogador da memória.

O responsável por este código é ?.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 10%.

## Read\_xpm

O ficheiro “read\_xpm.c” contém a função necessária para transformar um XPM num mapa de caracteres e saber a sua altura e largura.

O responsável por este código deverá ser João Cardoso, antigo professor de LCOM.

A participação e contribuição de Luís Barbosa é 0% e a de Carolina Centeio é 0%.

O peso relativo deste módulo no projecto é de 3%.

## RTC

## Sprite

“Sprite” é uma estrutura de dados que guarda informação sobre um XPM, nomeadamente posição-base no ecrã, largura, altura, velocidade de deslocamento e um apontador para um conjunto de caracteres.

O ficheiro “sprite.c” contém as funções necessárias para transformar um XPM num mapa de caracteres e saber a sua altura e largura, ou seja, criar um sprite, destruir um sprite, simular o movimento do sprite, desenhar um cursor, verificar colisões, desenhar um sprite, apagar do ecrã um sprite e criar o sprite “Dados”.

O responsável por este código deverá é ?.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 10%.

## Timer

O ficheiro “Timer.c” possui código para subscrever e des-subscrever interrupções e fazer uma contagem de tempo em que mais nenhum código pertencente à aplicação é executado.

O responsável por este código é Luís Barbosa.

A participação e contribuição de Luís Barbosa é X e a de Carolina Centeio é Y.

O peso relativo deste módulo no projecto é de 5%.

## UART

## Gráfico de chamada de funções

# Detalhes de implementação

Para implementar este jogo foram utilizados os conhecimentos já existentes e obtidos ao longo desta unidade curricular. També foi necessário pesquisar diversas outras fontes para implementar alguns labs, mas principalmente o UART. Ao longo do desenvolvimento do projecto procurou-se estruturar o código por camadas sendo que se calhar até se criaram mais camadas o que as necessárias. Também nos baseamos em programação orientada a objectos ao criarmos a estrutura de dados “Jogador”. Ao nível da placa gráfica trabalhamos num frame em memória auxiliar e quando este estiver pronto é enviado para a VRAM. Foi utilizado código assembly na implementação do RTC (pode-se colocar código assembly no teclado) falar da UART e RTC

Código baseado em eventos

Máquinas de estados

# Conclusões

Fornecer a estrutura de dados relativa às propriedades da placa gráfica para o LAB5

(Contribuição de cada um para o relatório)