**Faculdade de Engenharia da Universidade do Porto**



Block Breaker

Projeto Final

**Laboratório de Computadores 2020/21 – MIEIC:**

João Carlos Pascoal Faria

Pedro Alexandre Guimarães Lobo Ferreira Souto

Mário Miguel Fernandes Cordeiro

Estudantes & Autores:

Ângela Costa [up201907549@fe.up.pt](mailto:up@fe.up.pt) Patrícia Oliveira [up201905427@fe.up.pt](mailto:up201905427@fe.up.pt)

Resumo

O seguinte relatório foi realizado no âmbito da unidade curricular Laboratório de Computadores com o objetivo de apresentar o projeto final da mesma. Inclui etapas de implementação e componentes de interação com o utilizador, bem como explicitação de todos os dispositivos usados e especificação das suas tarefas no propósito do projeto.

Serve, de igual forma, como apoio para compreender a integração dos conteúdos lecionados e a participação de cada elemento na execução do projeto.

Palavras-Chave

palavra 1; palavra 2; ...

Índice

[Lista de figuras IV](#_Toc58880899)

[Lista de acrónimos V](#_Toc58880900)

[Glossário VI](#_Toc58880901)

[1. Introdução 1](#_Toc58880902)

[2. Instruções de utilização do programa 2](#_Toc58880903)

[2.1. Menu Principal 2](#_Toc58880904)

[2.2. Modo de jogo *Solo* 2](#_Toc58880905)

[2.2. Modo de jogo 1v1 3](#_Toc58880906)

[2.3 Menu Pausa 3](#_Toc58880907)

[2.4 *Leaderboard* 4](#_Toc58880908)

[3. Estado do projeto 5](#_Toc58880909)

[Lista das funcionalidades que nos propusemos inicialmente a implementar 5](#_Toc58880910)

[3.1 Funcionalidades não implementadas 5](#_Toc58880911)

[3.2 I/O Dispositivos usados 6](#_Toc58880912)

[2.1.1 Timer 6](#_Toc58880913)

[2.1.1 RTC 6](#_Toc58880914)

[2.1.2 Keyboard 7](#_Toc58880915)

[2.1.3 Mouse 7](#_Toc58880916)

[2.1.4 Video Card 7](#_Toc58880917)

[4. Organização/estrutura do código 9](#_Toc58880918)

[4.1 Módulos 9](#_Toc58880919)

[4.1 *Call graph* 9](#_Toc58880920)

[5. Detalhes de Implementação 10](#_Toc58880921)

[6. Conclusões 11](#_Toc58880922)

[Apêndice: Instruções de Instalação (opcional) 12](#_Toc58880923)

# Lista de figuras

Elemento a acrescentar

# Lista de acrónimos

[Elemento opcional.

Justifica-se se estes elementos (acrónimos, unidades, símbolos) ocorrerem com grande frequência no relatório. Quando ocorrerem pela primeira vez no texto deve apresentar-se a respetiva definição.

Para saber mais consulte o tutorial online “[Guia de Apoio à Publicação](https://docs.google.com/document/d/1TDC1behVq8x7fQL4CcPEEh_np5GXviJevQxnQ9gbiJs/edit#heading=h.6wf8v2wr6irp)”.]

# Glossário

[Elemento opcional.

Justifica-se sempre que seja necessário esclarecer o leitor sobre o significado de terminologia específica usada no texto no relatório.

Recomenda-se a sua localização nos elementos iniciais, embora na normalização existente haja variantes, podendo também constar nos elementos finais.

Para saber mais consulte o tutorial online “[Guia de Apoio à Publicação](https://docs.google.com/document/d/1TDC1behVq8x7fQL4CcPEEh_np5GXviJevQxnQ9gbiJs/edit#heading=h.6wf8v2wr6irp)”.]

# 1. Introdução

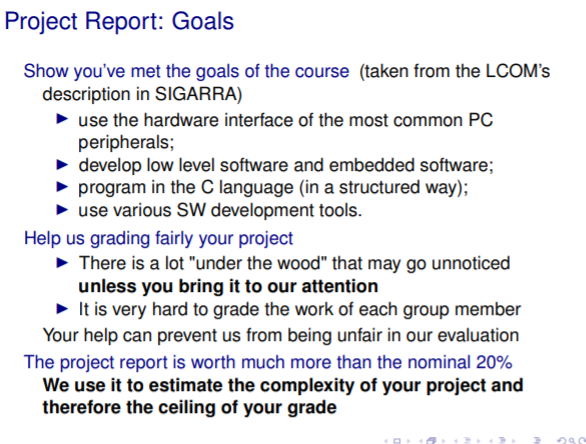
A aplicação desenvolvida no âmbito do projeto final da unidade curricular é um jogo baseado no original *Arkanoid*. Na nossa versão, o utilizador tem a possibilidade de jogar apenas um nível, com uma disposição de blocos já definida. O seu objetivo principal é destruir todos os blocos no menor tempo possível, sendo que um menor tempo leva a uma melhor pontuação.

Existe uma plataforma no final da tela de jogo que é movida horizontalmente, de forma a não deixar a bola passar abaixo desta. A bola quando entra em colisão com um bloco destrói-o. Entre intervalos definidos de tempo (neste caso de 30 em 30 segundos) o tamanho da plataforma diminui e a velocidade da bola aumenta até valores limite, com o propósito de gradualmente aumentar a dificuldade do nível.

Existe ainda outro modo de jogo do qual o utilizador poderá desfrutar, sendo ele o modo 1v1, onde 2 jogadores competem entre si, na mesma máquina, para limpar o cenário mais rápido que o seu oponente.

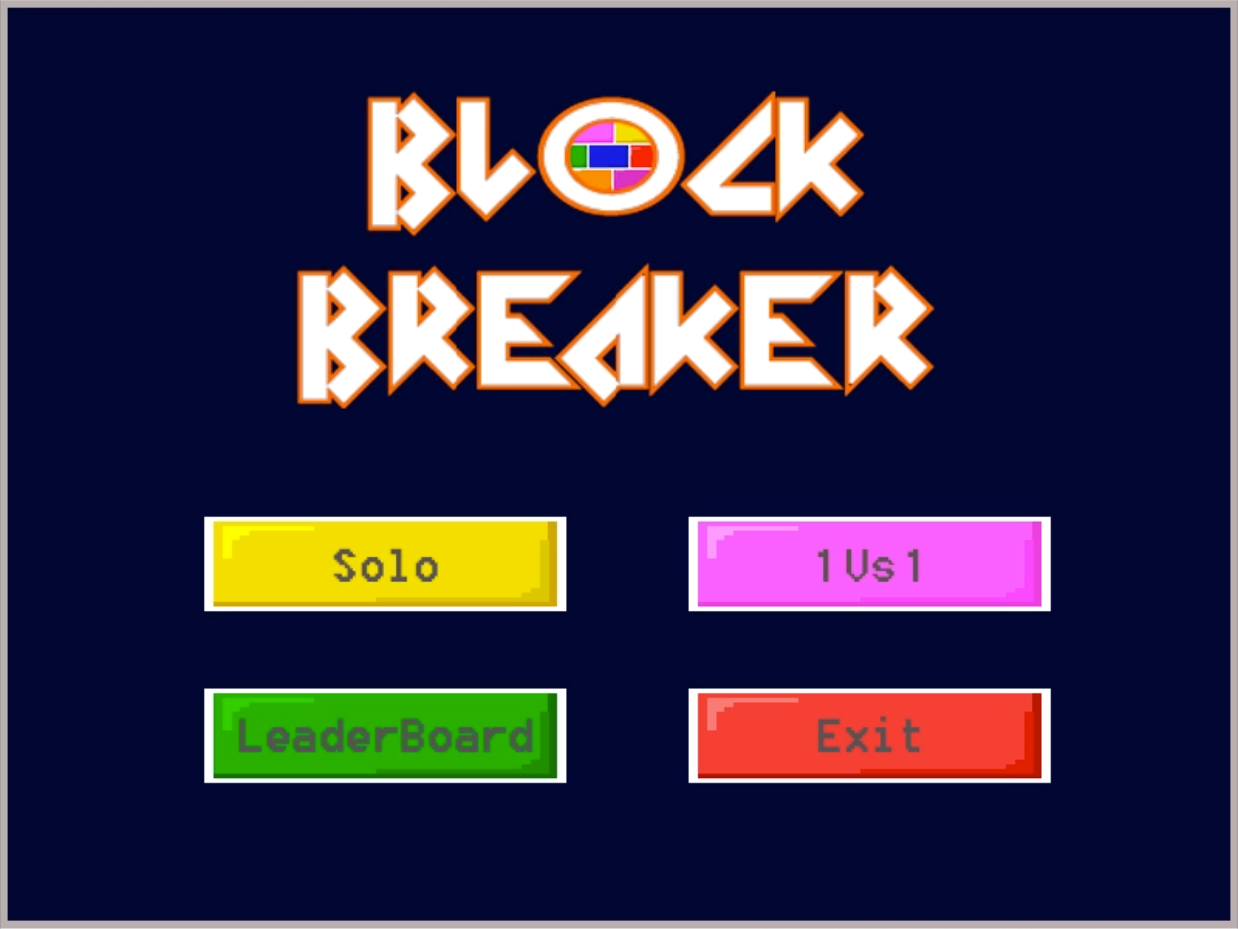
Durante o desenvolvimento do projeto foram considerados os diferentes objetivos da cadeira, implementando da melhor forma os requisitos mínimos do trabalho.

(apagar no final a imagem abaixo)



# 2. Instruções de utilização do programa

## 2.1. Menu Principal



Ao iniciar o programa é apresentado ao utilizador um menu inicial, a partir do qual o utilizador pode usar o rato para selecionar a opção desejada, entre escolher o modo de jogo, ver recordes ou sair da aplicação:

* Solo – inicia a aplicação em modo de jogo singular;
* 1v1 – inicia a aplicação em modo de jogo 1 contra 1;
* Leaderboard – apresenta uma lista dos 5 melhores jogadores;
* Exit – fecha a aplicação.

Ao movimentar o cursor do rato pelo ecrã sempre que este está sobreposto sobre uma possível opção a selecionar o cursor é alterado para o formato de apontador e o respetivo botão também muda de cor. Este comportamento é replicado para outros menus onde exista interação entre o rato e opções sobre a forma de botões.

Neste é igualmente apresentada a informação sobre o projeto, os autores e contexto, servindo, então como um ecrã de introdução da aplicação.

## 2.2. Modo de jogo *Solo*



Neste modo, o utilizador tem como objetivo limpar o cenário inicial (destruir todos os blocos) no menor tempo.

O cronómetro apenas começa a contar quando o utilizador interage com o jogo, ou seja, quando existe um movimento da plataforma. Para tal, o utilizador pode usar as teclas que correspondem às setas da direita e da esquerda no teclado ou como alternativa o conjunto de teclas ‘A’ e ‘D’. Ainda existe a possibilidade de o utilizador mover a plataforma através do movimento do seu rato, sendo este concordante com o anterior, desenhando no ecrã uma linha horizontal enquanto apenas o botão esquerdo se encontra premido.

Da mesma forma, a bola apenas iniciará o seu movimento quando existir a primeira interação com o utilizador, sendo esta projetada aleatoriamente em uma das possíveis direções. Sempre que uma bola é refletida pelos limites da tela de jogo ou pela colisão com um bloco após o destruir esta é lançada com o mesmo ângulo com que embateu nesse elemento de jogo.

São dadas ao utilizador 3 oportunidades para este terminar o nível, refletidas em vidas que são apresentadas no canto superior direito do espaço reservado para o cenário do jogo na forma de corações. O utilizador perde uma vida quando a bola passa para além da altura da plataforma (ultrapassa o ‘y’). Ao dar outra oportunidade ao jogador a bola e a plataforma são recolocadas nas suas posições inicial para continuar o nível.

Após o jogador terminar o nível (ficou sem vidas ou venceu-o) é lhe apresentado uma mensagem concordante com a forma de como o concluiu. Isto é, caso tenha ficado vem vidas é desenhado no ecrã “Game Over”, como demonstrado na imagem da esquerda. Caso tenha vencido o nível (destruiu todos os blocos) é desenhado “You Win” acompanhado da informação da respetiva pontuação que obteve, como podemos observar na imagem da direita. A pontuação de um jogador é calculada tendo em conta o tempo que demorou e o número de vidas com que terminou.



(imagem esquerda – game over) (imagem da direita you win)

As pontuações de todos os jogadores que concluíram com sucesso o jogo são guardadas de forma permanente num ficheiro, que contém a informação sobre o nome do jogador que realizou essa pontuação, bem como a data e hora em que esse jogo terminou.

## 2.2. Modo de jogo 1v1

(imagem do estado inicial do modo 1v1)

Neste modo, é replicado um mesmo nível 2 vezes no ecrã lado a lado. Cada lado corresponde a um jogador. O objetivo deste modo é limpar o cenário mais rápido que o oponente. Tal como no modo a ‘solo’ o cronómetro apenas inicia quando um dos jogadores interage com a respetiva plataforma e cada jogador tem começa com 3 vidas.

O jogador do lado esquerdo pode usar o conjunto de teclas ‘A’ e ‘D’ para mover a plataforma dentro dos limites do seu respetivo cenário, em quando o jogador do lado direito faz uso das setas da direita e esquerda no teclado. Em substituição do uso do teclado um jogador pode tirar partido do rato, usando este para movimentar a sua plataforma. Tal é possível para qualquer dos jogadores, ou seja, não se limita apenas a um lado do ecrã ou a um jogador específico.

No final do confronto, é apresentado ao jogador vencedor a mensagem “You Win” sobre a sua respetiva tela de jogo acompanhada da respetiva pontuação obtida (calculada pela mesma fórmula usada no modo a ‘solo’) e ao jogador vencido a mensagem “Game Over”.

(imagem de um confronto com as respetivas mensagens finais)

As pontuações obtidas neste modo não são contabilizadas para a *leaderboard* do jogo.

## 2.3 Menu Pausa



(substituir imagem do menu pausa – fundo de cor errada)

Em qualquer momento do jogo (quer no modo singular ou no modo com 2 jogadores) é possível pausar o jogo pressionando a tecla ‘ESC’. Neste menu são dadas as seguintes opções (a desejada é selecionada usando o rato, reconhecendo quando está sobre uma opção):

* Continuar o jogo – fecha o menu de pausa e retorna ao jogo;
* Reiniciar o jogo – reinicia de acordo com o modo de jogo onde se encontra (jogadores voltam a ter 3 vidas e o nível é desenhado no seu estado inicial);
* Desistir daquele jogo – volta ao menu inicial da aplicação.

Ao abrir este menu o jogo é pausado num estado de congelamento como se parasse no tempo, isto reflete-se no cronómetro que para de contar o tempo, na bola que para de se movimentar e na impossibilidade de alterar a posição da plataforma quer pelas teclas respetivas quer pelo rato.

## 2.4 *Leaderboard*



(substituir imagem de um exemplo de leaderboard)

O utilizador tem acesso à listagem dos melhores jogadores, ou seja, dos jogadores com as 5 pontuações mais altas obtidas no modo ‘solo’. Para cada jogador apresenta-se o seu nome e a respetiva pontuação, a data e hora em que o jogo correspondente foi realizado.

Para o desenho desta interface foi usada uma fonte de *xpms* concordante com o tipo de letra usado nos outros elementos do jogo já descritos.

Neste estado o utilizador pode optar por:

* Back – volta ao menu inicial da aplicação;
* Exit – fecha a aplicação.

(completar)

# 3. Estado do projeto

(a retirar esta 1 secção depois)

## Lista das funcionalidades que nos propusemos inicialmente a implementar

1. Desenho do cenário no seu estado inicial
2. Movimento da plataforma a partir das setas do teclado
3. Contador do tempo de jogo
4. Funcionalidades do jogo (rebentar os tijolos com a bola que é projetada pela barra)
5. Atribuir uma pontuação ao jogador de acordo com o tempo em que terminou o nível e quantidade de vidas
6. Permitir deslocar a plataforma também com o rato
7. Modo 1v1 (ganha quem limpar o cenário mais rápido): jogador 1 utiliza as teclas ‘A’ e ‘D’; jogador 2 as setas para a esquerda e direita; ou algum usa o rato em vez do par de setas correspondentes
8. Adicionar habilidades extra do jogo (velocidade da bola aumenta e a largura da barra diminui entre intervalos de tempo definidos)
9. Rato para a escolha das opções nos diversos menus do jogo (interação que deteta quando está sobre uma opção e altera a cor do botão e o cursor em concordância)
10. Menu principal inicial
11. Menu pausa
12. Guardar pontuações dos jogadores de modo solo de forma permanente em ficheiro (com informação da hora e data desse jogo) e apresentar um leaderboard no ecrã com os 5 melhores

(… se faltar alguma acrescentar)

## 3.1 Funcionalidades não implementadas

(passar para aqui as que não foram implementadas da lista acima)

## 3.2 I/O Dispositivos usados

A seguinte tabela identifica os dispositivos usados na realização do projeto e em que tarefas no âmbito específico das funcionalidades da aplicação.

|  |  |  |
| --- | --- | --- |
| Dispositivo | Função | Interrupções |
| *Timer* | Controlling frame rate; medição do tempo decorrido | S |
| *Keyboard* | Movimento da plataforma; pausar o jogo | S |
| *Mouse* | Navegação entre menus; movimento da plataforma | S |
| *Video card* | Menus da aplicação e exibição de telas | N |
| RTC | Obtenção da data e hora atual (registo de recordes) | ? |

Para cada uma das subsecções: (

describing the device’s functionality actually used;

referring to the code (function name) where you use it)

IMPORTANT Unless you mention the features you use in this section, you may not get credit for them I But remember, you must specify the names of the relevant functions

### 2.1.1 Timer

É utilizado como um cronómetro que identifica quando tempo passou desde o início do nível do jogo (inicia a contagem quando o jogador interage pela primeira vez com a plataforma na sua tela de jogo). De igual forma, serve para medir a passagem de tempo entre intervalos de tempo determinados previamente (neste caso de 30 em 30 segundos) para em concordância se alterar o tamanho da plataforma e a velocidade da bola até valores limite. Esta alteração é realizada 4 vezes. A cada decremento o comprimento da plataforma diminui em 12 pixéis, enquanto a velocidade da bola inicia a 1 e é incrementada em 1 unidade.

Controla, igualmente, o intervalo de tempo entre o qual o cursor do rato é desenhado em novas coordenadas.

Usado no seguinte código/função: …

(IMAGENS)

### 2.1.1 RTC

Para obter a data e hora atual, assim guardar com o registo da pontuação de um jogador no modo ‘solo’ a informação da altura em que essa mesma tentativa/partida foi realizada. Considerando que jogadores podem entrar com o mesmo nome na aplicação, uma vez que um mesmo jogador pode realizar mais do que uma tentativa estas são distinguidas pela sua data, uma vez que não é possível ocorrer dois jogos em simultâneo para propósitos de tentar registar um recorde.

Usado no seguinte código/função: …

(IMAGENS)

### 2.1.2 Keyboard

Usado apenas para propósitos do jogo. Controla o movimento da plataforma. Definindo os *makecodes* das teclas a usar no jogo, sempre que é gerada uma interrupção do teclado é verificado se esta tecla nos interessa e procede de acordo com a mesma.

Quando se pressiona a seta da direita ou ‘D’ a plataforma é movida para a direita em concordância com um deslocamento pré-definido. Quando se pressiona a seta da esquerda ou ‘A’ a plataforma é movida para a esquerda de acordo com o explicitado anteriormente. O movimento da plataforma apenas ocorre entre os limites direito e esquerdo da tela do jogo, uma vez que este é estritamente horizontal.

No entanto, pressionando-se a tecla ‘ESC’ durante um modo de jogo é aberto o menu de pausa da aplicação. Qualquer outra tecla é ignorada.

Usado no seguinte código/função: …

(IMAGENS)

### 2.1.3 Mouse

Usado o botão esquerdo do rato para selecionar a opção desejada nos diferentes menus da aplicação (é calculado se o rato se encontra numa coordenada válida [região no ecrã que corresponde a uma opção] e qual a opção escolhida de acordo com essa respetiva coordenada, mudando esta de cor quando o rato se encontra sobre ela).

Como outro mecanismo que permite a movimentação da plataforma dentro do limite da tela do jogo. Para tal é necessário premir continuamente o botão esquerdo enquanto se desenha uma linha horizontal que representa o desejado deslocamento, dentro de uma definida tolerância vertical. Esse movimento do rato é replicado pelo movimento da plataforma. Se durante esse desenho o jogador pressionar outro qualquer botão este invalida o movimento da plataforma, pelo que esta parará na posição respetiva onde esse mesmo botão foi pressionado. Da mesma forma, que quando levantar o botão esquerdo ou quando chegar ao limite do cenário essa mesma situação acontecerá, a plataforma parará de se mover.

Ao iniciar as interfaces onde é necessário o uso e movimento do rato este é desenhado inicialmente perto do centro da mesma, em coordenadas pré-definidas (x: 400, y: 350).

Usado no seguinte código/função: …

(IMAGENS)

### 2.1.4 Video Card

Numa fase inicial desenhamos o nosso jogo apenas para o modo gráfico direto 0x115 (resolução de 800x600, modo de cor direta com 24 (8:8:8) bits por pixel em formato R:G:B), uma vez que consideramos que as dimensões deste eram apropriadas tendo em conta a versão original do jogo onde nos baseamos. Por outro lado, também estávamos a ter dificuldades em desenhar no ecrã uma *xpm* que não fosse indexada. Pelo que nos focamos apenas num modo e resolvemos o problema para este, interpretando a construção e organização da cor de um pixel na *struct xpm\_map\_t* para este modo específico (neste caso, como a cor é guardada em formato 8:8:8 tivemos que 3 de 3 em 3 bytes e assim construir a cor em formato RGB).

(imagem da função)

Todas as componentes gráficas usadas no jogo foram desenhadas por nós, como por exemplo: os blocos de diferentes cores, as plataformas com diferentes tamanhos, o coração que representa uma vida, a bola, os números que constituem o relógio do jogo e todos os títulos dos menus apresentados e suas opções. Foi criado um rascunho da disposição do jogo e diferentes elementos, analisadas as dimensões necessárias para cada constituinte principal e depois feitos estes individualmente. Estes desenhos foram realizados no Photoshop, guardados em formato *png* e finalmente convertidos para *xpm* respetiva com recurso à ferramenta GIMP, formato no qual são usados no projeto [tendo sempre por base o modo gráfico 115].

(imagem do rascunho com a anotação de dimensões e diferentes componentes)

Desenho de funções gerais de desenho, uma vez que para uma tela de jogo definimos que esta seria desenhada tendo em conta as coordenadas de referência para a mesma (canto superior esquerdo). Para tal para um objeto que é desenhado é guardada a distância a que este se encontra em relação às coordenadas do canto (verticalmente e horizontalmente), permitindo assim, futuramente desenhar “transportar” toda a tela de jogo para outras coordenadas modificando apenas estas coordenadas (x, y) de referência. Facilitando igualmente a replicação de uma tela de jogo num mesmo ecrã em 2 posições diferentes lado a lado, como demonstrado no modo 1v1.

(imagem de distancias em relação ao canto)

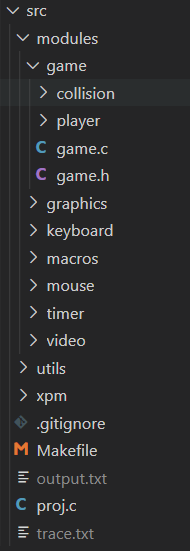
Relacionado com a placa de vídeo no projeto podemos ainda identificar outras componentes que foram implementadas/usadas:

* Carregamento das xpm’s de todos os elementos do jogo no início da execução da aplicação para eliminar o *flickering* originado pelo desenho sucessivo de um elemento em posições diferentes;
* *Double buffering*: associado ao movimento do rato, com o objetivo de voltar a desenhar no ecrã o que se encontrava sob o rato antes da modificação da sua posição. Esta operação é realizada sempre que se interage com o rato modificando as coordenadas do desenho do seu cursor;
* Movimento de objetos: bola movimenta-se dentro dos limites da tela do jogo e sempre que é detetada uma colisão com um bloco esta destrói-o, mas no caso de colidir com a plataforma esta é projetada com a mesma inclinação com que embateu na plataforma, mas para o lado oposto; as opções apresentadas nos menus têm uma pequena “animação” associada – sempre que se coloca o rato sobre o botão que representa o espaço ocupado por essa opção é alterada a cor deste, bem como a imagem do cursor também se altera;
* Fontes de letras: para escrever todos os dados na interface da *leaderboard*, desde o nome dos jogadores, data e hora e pontuações, em concordância com as fontes usadas noutros elementos da aplicação.
* Funções VBE (exemplo: alterar a palete de cores; *page flipping*): …

Usado no seguinte código/função: …

(IMAGENS)

# 4. Organização/estrutura do código

O código foi modularizado e distribuído por diferentes ficheiros agrupados por pastas, que intuitivamente permitem perceber a função de cada ficheiro no projeto e encontrar uma determinada parte do código mais facilmente.

Para tal foi imperativo adicionar ao *Makefile* a seguinte diretiva:



O excerto acima, faz com que os executáveis de cada ficheiro sejam gerados na raiz do diretório (onde se encontra o *Makefile*), eliminando os problemas de inclusões de ficheiros entre pastas diferentes.

No desenvolvimento do projeto foi reaproveitado todo o código já realizado ao longo dos laboratórios que dizem respeito aos dispositivos usados.

## 4.1 Módulos

### Timer

- um paragrafo de descrisao que inclui um peso medio

- informaça de quem vez o que aqui

- main structures of the module

### Teclado

- um paragrafo de descrisao que inclui um peso medio

- informaça de quem vez o que aqui

- main structures of the module

### Rato

- um paragrafo de descrisao que inclui um peso medio

- informaça de quem vez o que aqui

- main structures of the module

### Vídeo

- um paragrafo de descrisao que inclui um peso medio

- informaça de quem vez o que aqui

- main structures of the module

### Gráficos

- um paragrafo de descrisao que inclui um peso medio

- informaça de quem vez o que aqui

- main structures of the module

### Game

- um paragrafo de descrisao que inclui um peso medio

- informaça de quem vez o que aqui

- main structures of the module

### Player

- um paragrafo de descrisao que inclui um peso medio

- informaça de quem vez o que aqui

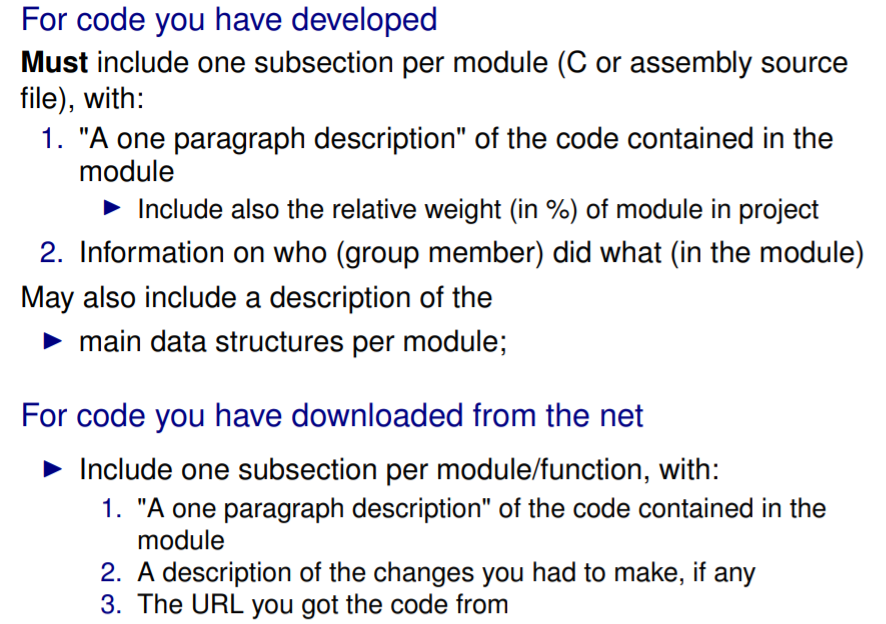
- main structures of the module

### Subsection

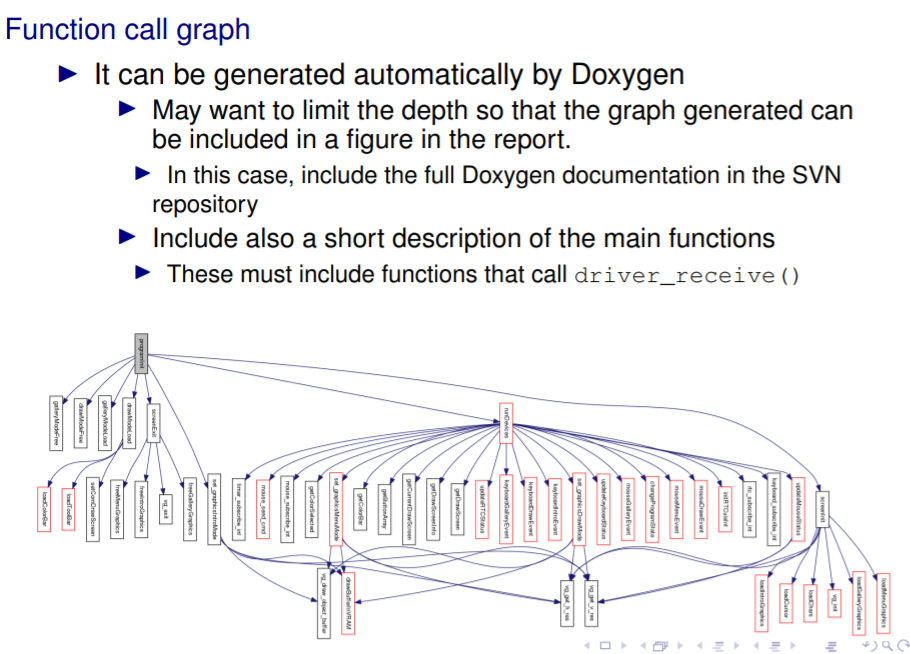
- um paragrafo de descrisao que inclui um peso medio

- informaça de quem vez o que aqui

- main structures of the module



## 4.2 *Call graph*



# 5. Detalhes de Implementação

(trocar a ordem para fazer mais sentido – como crescimento gradual)

## 5.1 Máquina de estados do desenho de uma linha horizontal

De forma a detetar quando uma linha horizontal é corretamente desenhada enquanto o botão esquerdo do rato está pressionado foi implementada a seguinte máquina de estados, representada pela tabela das funções de transição da mesma.

|  |  |  |  |
| --- | --- | --- | --- |
| **Estado Atual** | **Evento** | **Próximo Estado** | **Output** |
| INIT (Inicial) | LB\_PRESSED | DRAW |  |
| DRAW (Desenho) | LB\_RELEASED | COMP | Terminar movimento |
| DRAW (Desenho) | BUTTON\_EV | INIT | Reiniciar movimento |
| DRAW (Desenho) | MOUSE\_MOV | DRAW | Desenhar no novo conjunto de coordenadas (*True*) |
| DRAW (Desenho) | HOR\_TOLERANCE | INIT | Reiniciar movimento |
| COMP (Completo) | LB\_PRESSED | DRAW |  |

Legenda:

LB\_PRESSED: Apenas o botão esquerdo do rato premido;

LB\_RELEASED: Botão esquerdo do rato largado;

BUTTON\_EV: Evento gerado pela interação com outro botão que não o esquerdo;

MOUSE\_MOV: Existe movimento do rato;

HOR\_TOLERANCE: Tolerância vertical excedida para o desenho de uma linha horizontal.

É alterado um estado da máquina considerado como *static* de acordo com um determinado evento. A deteção deste é realizada por uma função externa ao processamento da máquina, também implementada por nós.

(incluir recorte da função)

## 5.2 Função de movimento da bola

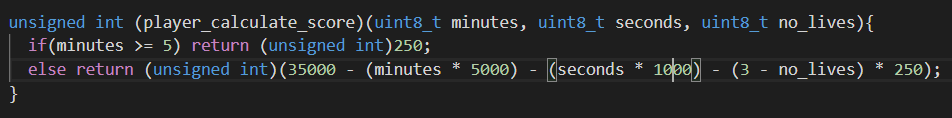
Função de movimento da bola que define como esta à refletida (explicação da lógica de pensamento, etc)

## 5.3 Deteção de colisões durante o jogo

Forma que são detetadas colisões (da bola com a plataforma e com os blocos)

## 5.4 Fórmula de cálculo da pontuação de um jogador

A pontuação de um jogador diminui proporcionalmente com o tempo que demora a concluir o nível e o número de vidas com que o termina com sucesso, desde um valor máximo pré-definido suficientemente grande, para que deste cálculo não resultem pontuações com valores negativos. Até a um mínimo constante (quando ultrapassa os 5 minutos de jogo). A expressão aplicada foi a seguinte:

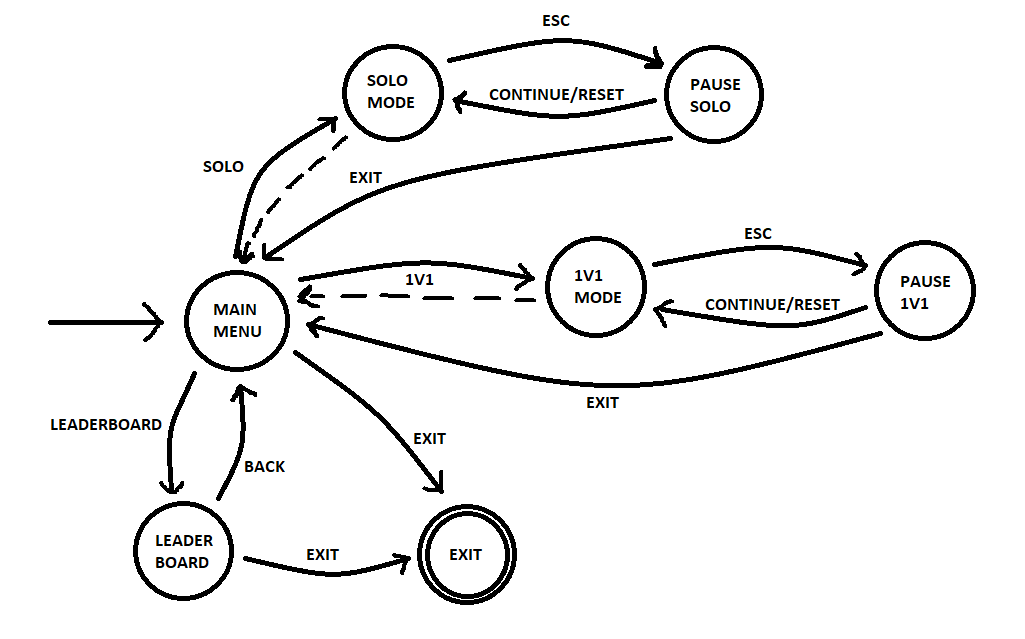


## 5.5 Implementação da função *vídeo\_mode\_get\_config()*

De forma a não usar a função importada pela lcf, implementamos a nossa própria função…

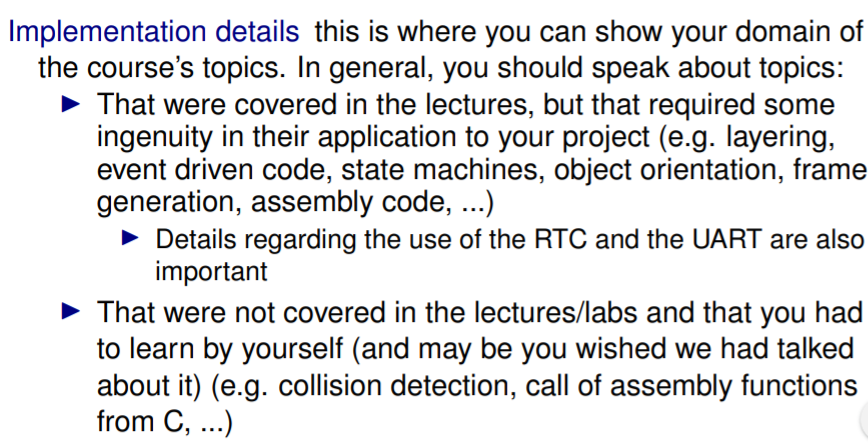
## 5.6 Máquina de estados da navegação entre menus

De forma a usar apenas um ciclo *driver\_received()* e ainda assim ter a informação do estado em que a aplicação se encontra, ou seja, em que interface o utilizador está, apresentado as informações necessárias e tratando as interrupções em cada um dos processos em concordância, foi idealizada e implementada uma máquina de estados que representa essa esta navegação entre menus. Controlando estas transições de acordo com eventos gerados pelo utilizador (seleção de opções ou pressionar a tecla “ESC” durante um modo de jogo) sabemos sempre o que apresentar e como reagir aos mesmos de forma mais automatizadas.

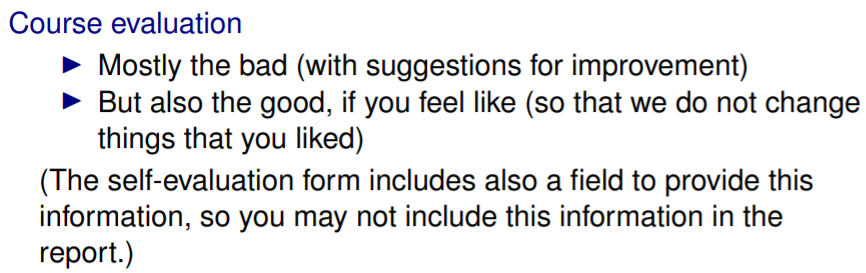


Existe uma transação implícita dos modos de jogo para o menu principal, quando o utilizador termina um jogo (vencendo ou perdendo), mas esta não deriva de um evento produzido pela interação direta do utilizador com a aplicação, pelo que não é explícito no diagrama da máquina de estados.

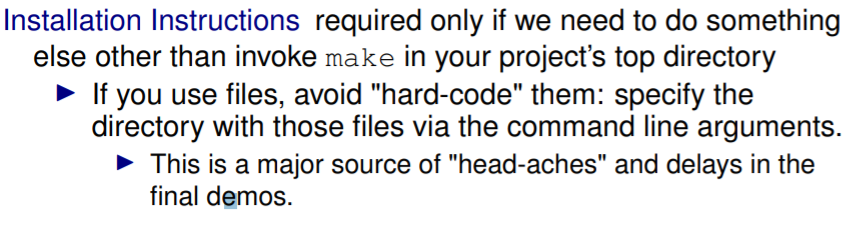
(acrescentar mais tópicos se considerado necessário)



# 6. Conclusões



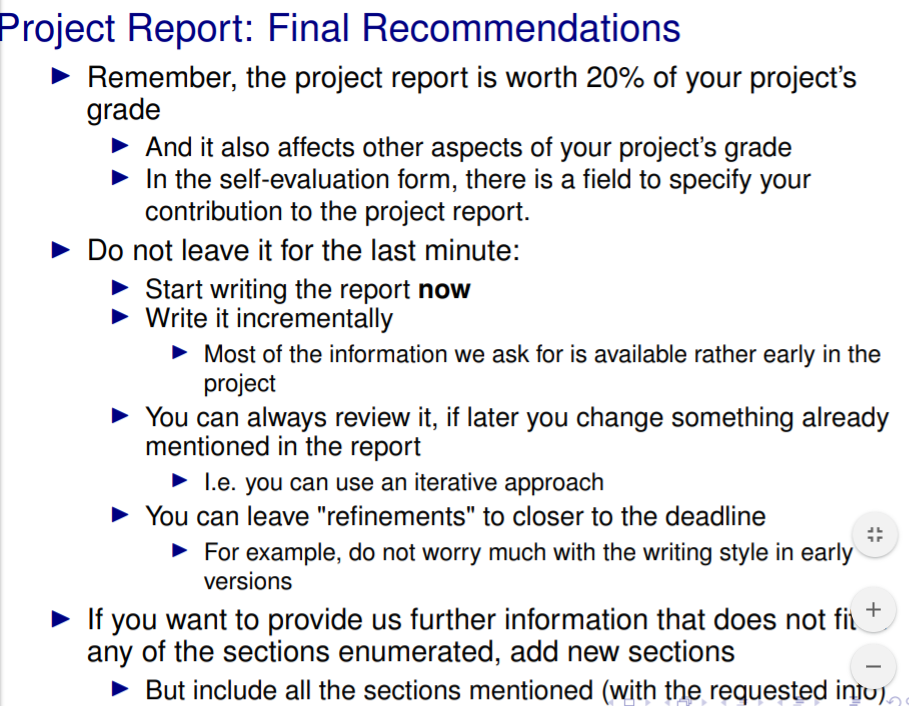
# Apêndice: Instruções de Instalação (opcional)



Para executar a aplicação é necessário informar o modo gráfico a usar (apenas para 115) e opcionalmente o nome do jogador que tentará concluir o nível, para guardar esta informação com a respetiva pontuação obtida. Caso nenhum nome seja introduzido, por defeito, este será considerado como *anonymous*.

Se as regras apresentadas acima não forem compridas será apresentada a seguinte mensagem:

(introduzir print\_usage)



(manter o seguinte como exemplo)

Equacoes:

(1)

* as figuras devem ser legendadas por baixo;
* as tabelas devem ser legendadas no topo.



Figura 1. Fotografia aérea do Campus da FEUP,   
campus da FEUP realçado a cor

Tabela 1: Recursos Físicos da FEUP (excerto adaptado de “A FEUP em números”, 2011)

|  |  |
| --- | --- |
| Área total do campus FEUP | 93 918 m2 |
| Espaços verdes | 23 000 m2 |
| Número de computadores dedicados ao ensino | 1 815 |
| Investimento em equipamentos de laboratório | 1,46 M€ |