

Penguin Run

Relatório de Laboratórios de Computadores

João Afonso Martins Domingues Andrade (up201905589)

Sérgio Miguel Rosa Estêvão (up201905680)

Dezembro 2020 - Janeiro 2021

Conteúdo

1	Introdução	3
1.1	Main Menu	3
1.2	Menu SinglePlayer	4
1.3	Nível 1	5
1.4	Menu de Pausa	6
1.5	Nível 2	8
2	Periféricos utilizados no Projeto	10
2.1	Timer	10
2.2	Keyboard	10
2.3	Mouse	11
2.4	Video Card	11
2.5	RTC	12
2.6	UART	12
3	Organização e estrutura do código	13
3.1	Módulo graphics	13
3.2	Módulo 8042	13
3.3	Módulo 8254	13
3.4	Módulo keyboard	13
3.5	Módulo menu	14
3.6	Módulo mouse	14
3.7	Módulo movement	14
3.8	Módulo RTC	14
3.9	Módulo sprite	15
3.10	Módulo timer	15
3.11	Peso relativo dos módulos	15
4	Function call graph	17
5	Detalhes da Implementação	18
5.1	Object Oriented Language	18
5.2	Best Times e Files	18
5.3	Sprites (XPM)	18
5.4	Colisões	19
5.5	Layering	19
5.6	Uso dos frames	19
5.7	Doxygen	19
6	Conclusão	20

1 Introdução

De acordo com os objetivos da cadeira de LCOM, decidimos realizar uma versão remastered de certa forma do jogo Frogger, à qual chamamos de Penguin Run. O objetivo do jogo é conseguir ultrapassar uma série de obstáculos de forma a conseguir chegar à meta que se encontra no lado oposto do ecrã. Este projeto teve também como objetivo conseguirmos usar os mais variados periféricos do computador.



Fig 1 - Menu Inicial

1.1 Main Menu

No menu, existem duas opções: Singleplayer e Quit.

Na opção de Singleplayer, irá ser apresentado o menu de singleplayer que constitui os níveis que se encontram disponíveis.

Na opção Quit o programa irá terminar.

Para escolher uma das opções, basta passarmos por cima das opções com o rato, sendo bastante perceptível pois a opção fica mais carregada.

1.2 Menu SinglePlayer



Fig 2 - Menu SinglePlayer

Neste menu, são-nos apresentados sete opções de botões, três que correspondem aos níveis disponíveis, outros três que correspondem aos recordes de tempos em cada nível respetivamente e outro que nos permite voltar ao Main Menu. Ao clicarmos em "Best Times" irá ser apresentada uma janela igual á seguinte:



Fig 3 - Best Times

Nesta janela é nos apresentado, como já foi dito antes, os três melhores tempos de cada nível, existindo o 1º, 2º e 3º lugar, havendo ainda informação extra sobre o dia, mês e ano em que esse recorde foi realizado.

1.3 Nível 1

Ao escolhermos o nível 1, irá-nos ser apresentada a seguinte janela dando início ao jogo:



Fig 4 - Nível 1

Quanto às regras e explicação do jogo, podemos desde logo à partida perceber a existência de diversos obstáculos, entre os quais, duas paredes de gelo que se deslocam horizontalmente com um speed definido, duas paredes de pedra que impedem o avanço do pinguim e são imóveis, um boneco de neve imóvel também e um tronco que se desloca horizontalmente e representa a única maneira que o pinguim tem de atravessar o rio para o outro lado sem morrer.

O embate com qualquer um dos objetos, à exceção do tronco, provoca a morte do pinguim, o que significa que o objetivo do mesmo é alcançar o topo do ecrã, ultrapassando cada obstáculo á medida que vai andando para cima. Quando chega ao rio, o mesmo irá ter que esperar pelo tronco, sendo que uma vez em cima do tronco se desloca com o mesmo, tendo porém de sair antes de chegar

ao limite da direita do ecrã.

Também é possível vermos o cronómetro de nível que nos indica o tempo passado desde o início do nível, tempo este que se melhor que algum dos apresentados em Best Times, substitui o mesmo.

1.4 Menu de Pausa

Durante o jogo, o jogador pode decidir pressionar o botão "esc", o que provoca uma pausa não só no jogo, como também no cronómetro do mesmo. A janela de pausa apresentada é a seguinte:

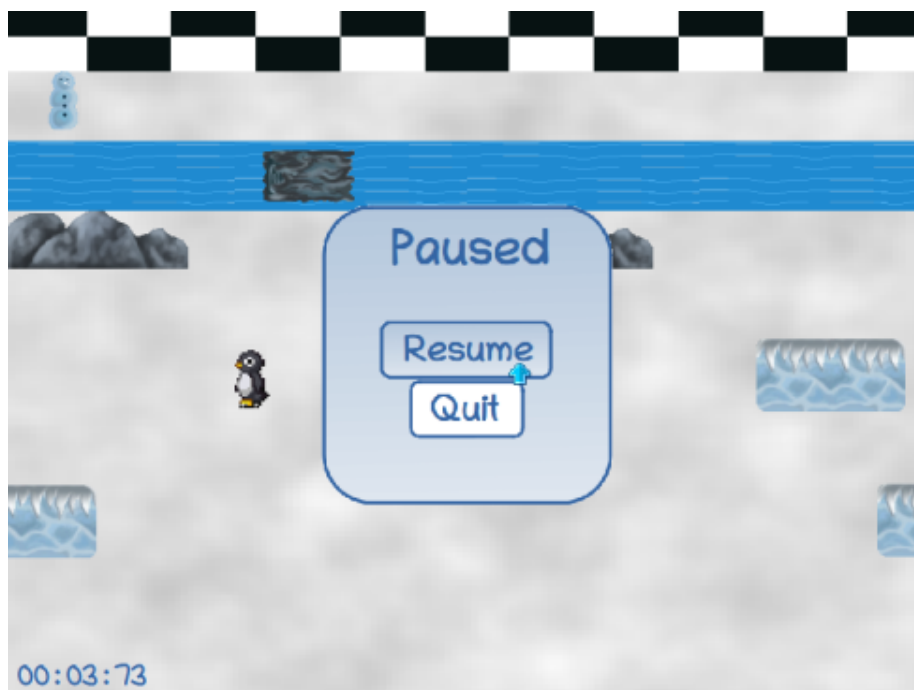


Fig 5 - Menu de Pausa

Nesta janela nova, temos duas opções sendo elas "Quit" que nos faz retornar ao menu SinglePlayer e "Resume" que continua o jogo, retomando novamente o cronómetro também.

Se o jogador embater em algum dos obstáculos, morrerá e aparecerá a seguinte janela:

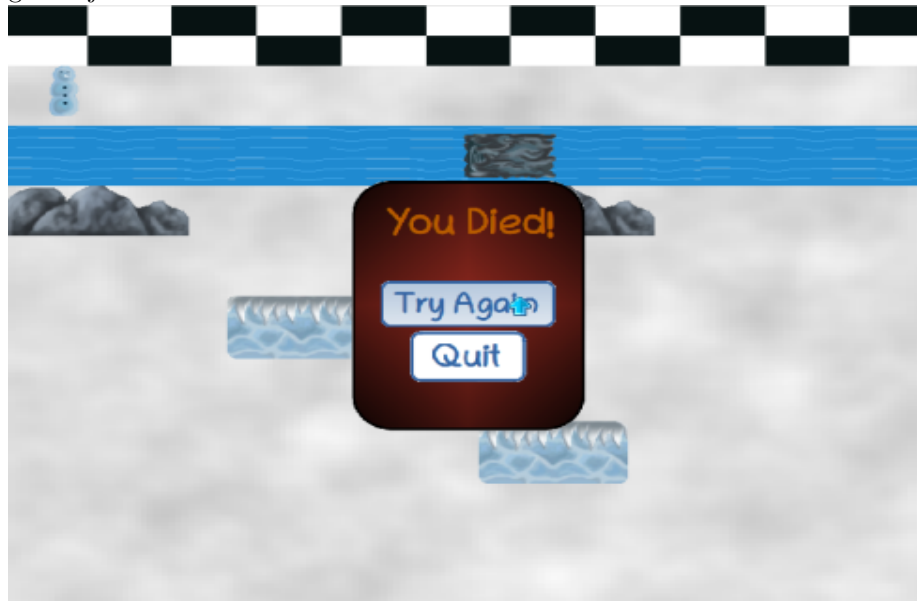


Fig 5 - Menu de Morte

Que permite o jogador tentar outra vez, no botão "Try Again", ou sair para o menu SinglePlayer, no botão "Quit".



Fig 6 - Menu de Victória

No menu de vitória é nos apresentado o tempo total gasto pelo jogador e um botão que nos permite voltar ao menu SinglePlayer.

1.5 Nível 2

O segundo nível, segue as mesmas regras que o primeiro, havendo apenas distinção nos obstáculos e posicionamento dos mesmo.

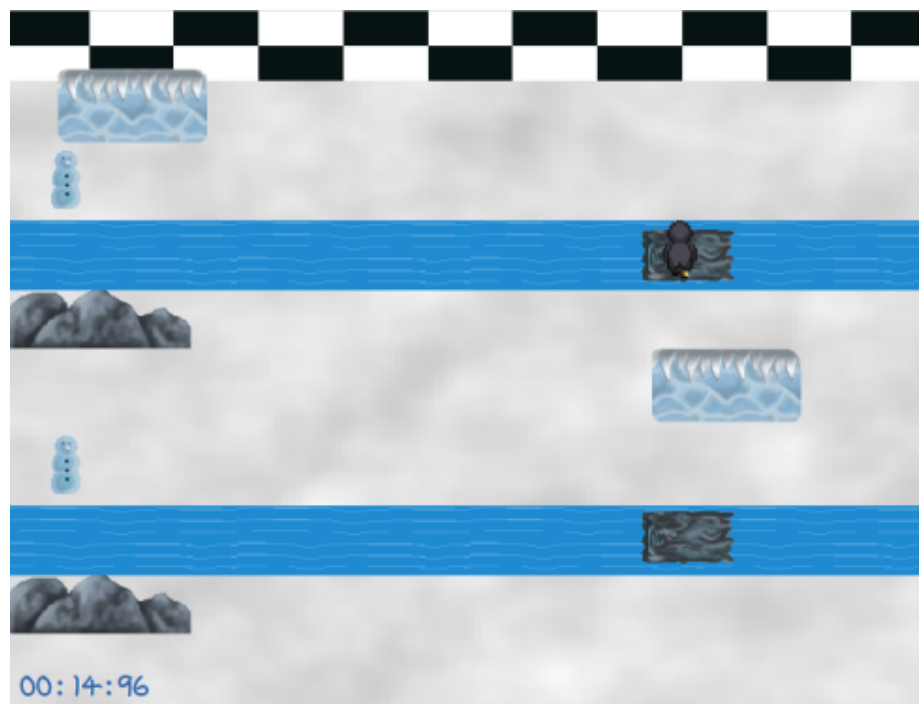


Fig 7 - Nível 2

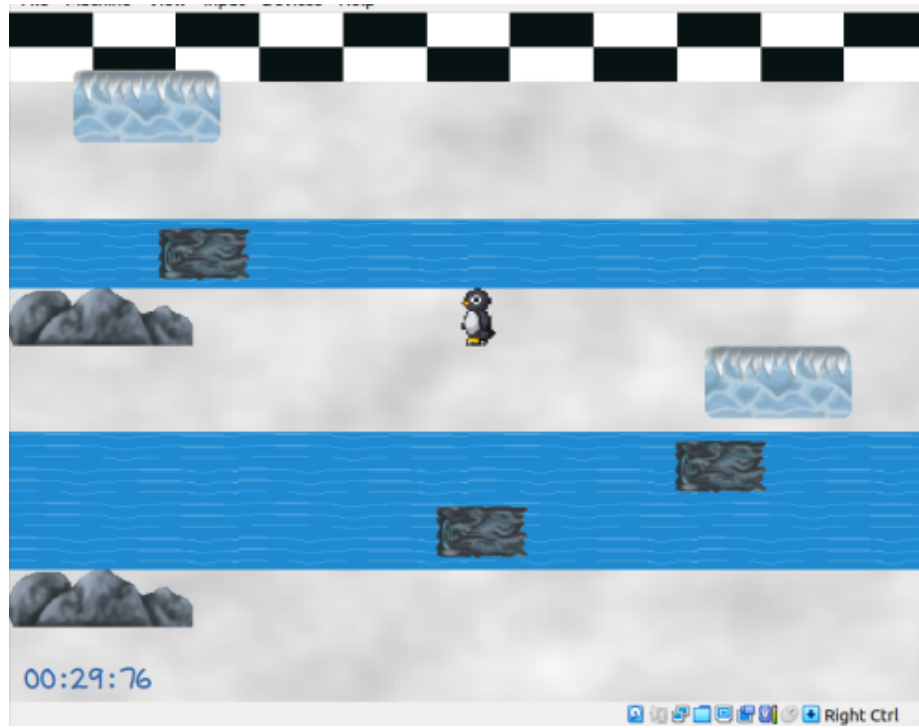


Fig 8 - Nível 3

2 Periféricos utilizados no Projeto

Periférico	Função	Interrupções
Timer	Utilizado no cronómetro e no controlo da cadência de frames	sim
Keyboard	Controlo do movimento do pinguim	sim
Mouse	Usado para escolher opções em todos os menus	sim
Video Card	Apresentação do ecrã e de cada um dos seus componentes	sim
RTC	Apresentação da data de cada um dos recordes de tempo	sim
UART	Não foi implementado	não

2.1 Timer

O timer foi utilizado neste projeto, principalmente para controlar a cadência dos frames tendo também funcionalidade extra no uso do cronómetro de jogo. Para tal recorremos a algumas funções utilizadas no lab2, entre as quais, o `timer_subscribe_int()` e a `timer_unsubscribe_int()`

Para conseguirmos alterar a cadência dos frames, utilizamos a função `timer_set_frequency()` que altera a frequência do timer 0.

A frequência escolhida por nós foi de 30 Hz pois foi a que proporcionou a melhor experiência de jogo.

2.2 Keyboard

No nosso projeto, o keyboard foi principalmente utilizado na movimentação do pinguim e também de forma colocar o jogo em pausa.

Assim, é permitido ao jogador utilizar tantas as teclas 'w', 'a', 's', 'd' como também as setas para se movimentar, sendo 'w' e a seta para cima o movimento para cima, 's' e seta para baixo o movimento para baixo, 'a' e seta para a esquerda o movimento para a esquerda e por fim 'd' e seta para a direita o movimento para a direita.

Para pausar o jogo, basta ao jogador precionar a tecla 'esc'.

Assim como o timer foram também utilizadas algumas funções do lab3 tais como o `keyboard_subscribe_int()` e o `keyboard_unsubscribe_int()`.

2.3 Mouse

No nosso projeto, este periférico foi usado principalmente na navegação entre todos os menus.

Assim é permitido ao jogador mexer o rato ao longo de todo o ecrã sendo que, quando está por cima de um botão de um menu o mesmo dá uma espécie de pop-up para se destacar.

O jogador pode depois escolher clicar no botão do lado direito do rato, escolhendo essa opção.

Foram também usadas algumas funções do lab4 tais como, o `mouse_subscribe_int()`, e o `mouse_unsubscribe_int()`.

2.4 Video Card

A video card foi utilizada neste projeto e permitiu-nos desenvolver toda a parte gráfica do mesmo, ou seja, foi usada no desenho de todos os menus, obstáculos, pinguim, botões e números ou palavras, sendo que foi definida uma função para desenhar cada um destes componentes.

Quanto ao modo que escolhemos usar no nosso projeto foi o modo VBE 0x115 que nos dá acesso a um ecrã de 800 pixels de largura e 600 de altura, contendo 24 bits por cada pixel.

O movimento dos obstáculos (tais como as paredes de gelo e o tronco) é possível graças a este periférico que nos permite através de uma função que dá update à posição dos mesmo, desenhá-los nas suas sucessivas posições diferentes, assim como no movimento do pinguim.

Utilizamos também double buffering que tornou o nosso jogo de certa maneira muito mais "suave" levando também a uma melhor experiência de jogo.

Utilizamos também algumas funções do Lab5 tais como, a `vge_init()`, `vge_exit()` e a `vbe_get_mode_info()`, que nos ajudou a definir o modo que pretendíamos.

2.5 RTC

O RTC teve um papel importante no nosso projeto pois permitiu-nos guardar a data, sendo esta constituída por dia, mês e ano, em que cada um dos tempos recordes presentes no "Best Times" foi atingido.

Utilizamos ent uma função para verificar se o 'RTC' está em update ou não (pois não é possível a leitura do mesmo quando em update), outra para ir buscar aos registos do RTC o dia, mes e ano e outra por fim par converter a data de BCD para binário.

Imagem de exemplo do RTC onde podemos ver a data (03/01/21) e o tempo correspondente:



Fig 9 - RTC

2.6 UART

Apesar de termos algumas ideias que gostávamos de experimentar no que toca a este periférico, não tivemos oportunidade para tal.

3 Organização e estrutura do código

3.1 Módulo graphics

O módulo graphics contém todas as funções relacionadas com todo o design gráfico.

Assim, neste módulo encontram-se as funções de `change_color()` que muda a cor de cada pixel, a `vg_draw_pixmap()` que desenha um pixmap passado como parâmetro, e diversas funções para desenharmos tanto os obstáculos, como as datas, tempos recorde, pinguim e algumas letras utilizadas nos menus (`vg_draw_Penguin()`, `vg_draw_buttons()`, `vg_draw_int()` etc..).

Possuí também as funções de `clear_screen()` e `double_buffer()` que fazem a manutenção do ecrã e aplicam o double buffering, respetivamente.

Estas funções foram desenvolvidas de igual modo pelos dois membros do grupo.

3.2 Módulo 8042

Este módulo foi-nos útil pois continha maior parte das constantes que necessitamos para corretamente implementar o rato e teclado.

Foi-nos fornecido durante a realização dos Labs 3 e 4.

3.3 Módulo 8254

Este módulo foi-nos útil pois continha maior parte das constantes que necessitamos para corretamente implementar o timer.

Foi-nos fornecido durante a realização do Lab 2.

3.4 Módulo keyboard

Neste módulo estão presentes todas as funções que estão relacionadas com o uso do teclado tais como a `keyboard_subscribe_int()` que subscrive os interrupts do keyboard e `kbc_ih()` que trata das interrupções etc..

Maior parte das funções foram provenientes da realização do Lab3, tendo sido feitos alguns ajustes pelos dois membros do grupo.

3.5 Módulo menu

Neste módulo estão presentes todas as funções relacionadas com os menus, botões presentes nos menus, construção do jogo e também todas as structs relacionadas etc...

Também estão presentes neste módulo todas as funções relacionadas com o update dos arrays que contém os recordes de tempo, e também os arrays que contém as datas correspondentes (Change_Menu(), AssignButton() etc...).

Relacionado com os melhores tempos e datas, estão também as funções de escrita e leitura para ficheiros dos tempos e datas (saveScorestoFile(), updateBestTimes() etc...).

Neste módulo as funções relacionadas com toda a logística dos menus e mouse foram feitas pelo Sérgio Estêvão, e a parte dos ficheiros e update dos tempos recorde pelo João Afonso.

3.6 Módulo mouse

Neste módulo foram utilizadas maior parte das funções do lab4, e foram acrescentadas algumas tais como : update_mouse() etc...)

Contém também as funções de subscribe e unsubscribe dos interrupts do mouse (mouse_subscribe_int(), e o mouse_unsubscribe_int()) e de tratamento de interrupts (mouse_ih()).

As funções que foram acrescentadas e o tratamento foi realizado por parte do Sérgio Estêvão.

3.7 Módulo movement

Neste módulo está presente a função principal do jogo (game()), onde maior parte das funções serão chamadas, sendo esta a função que inicia o jogo e que faz a manutenção do mesmo.

Este módulo foi realizado por ambos os membros do grupo.

3.8 Módulo RTC

Neste módulo encontram-se todas as funções relacionadas com o funcionamento do RTC tais como(date() que retorna a data atual etc...)

Está também presentes funções de verificação de updates do RTC e conversão da data de BCD para binário.

Este módulo foi realizado por parte do João Afonso.

3.9 Módulo sprite

Neste módulo estão presentes funções relacionadas tanto com o pinguim (`build_penguin()`, `resetpenguin()` etc...) como com todos os objetos presentes no jogo (`buidlObject()`, `update_objects()` etc...).

Também estão presentes neste módulo todas as funções relacionadas com a verificação de vitória (`checkWin()`), colisões (`checkCollisions()`), contacto com a água (`checkInWater()`), contacto com o rio etc...

Este módulo possui também todas as structs relacionadas com os objetos e o pinguim e também algumas enums relacionadas com os mesmos.

Neste módulo, a logística dos objetos, colisões e verificações foram feitas pelo João Afonso e a logística do pinguim, update do mesmo e reset do mesmo foi feito pelo Sérgio Estêvão.

3.10 Módulo timer

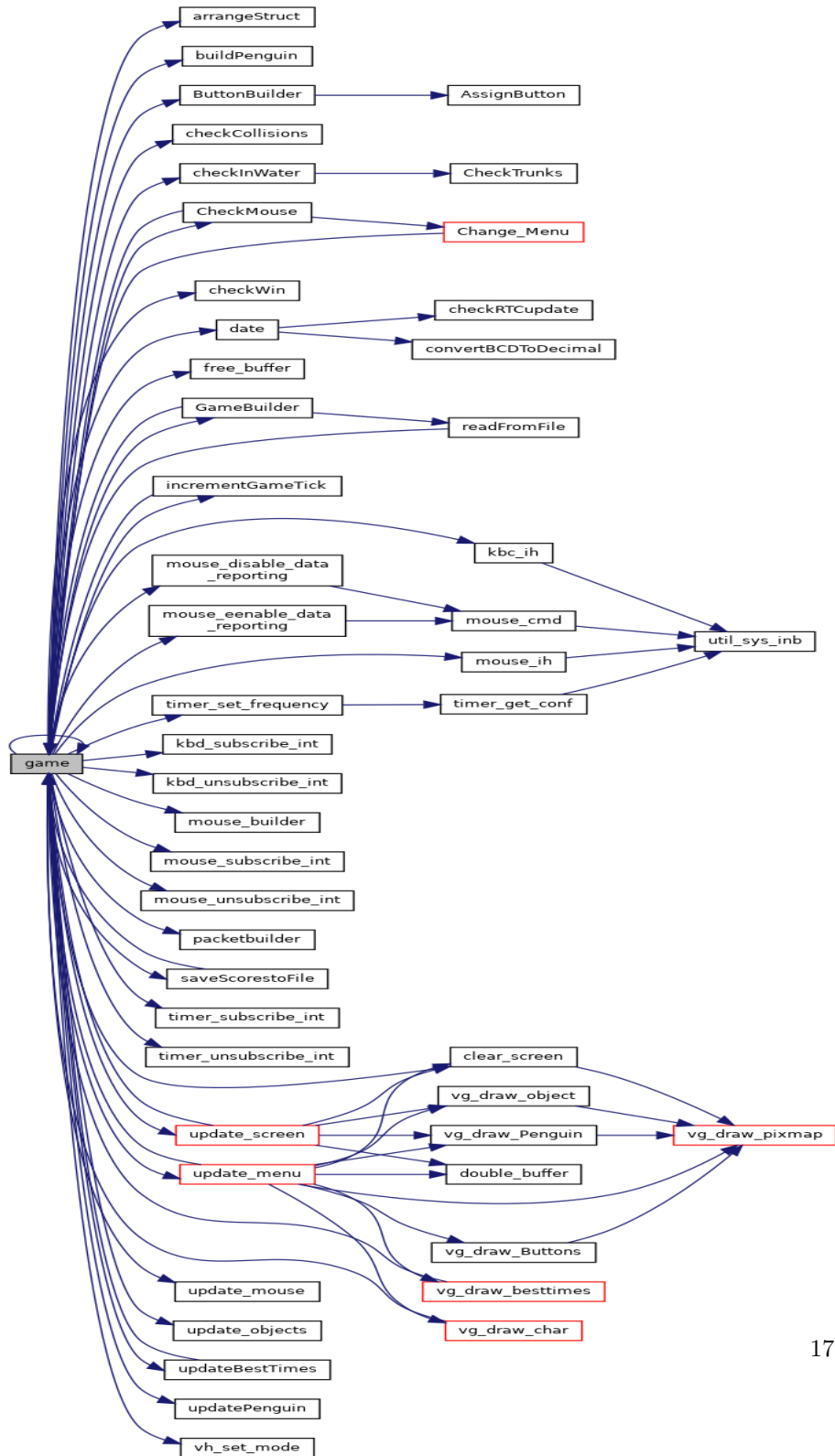
Este módulo foi desenvolvido na totalidade na realização do Lab2.

Contém as funções de mudança de frequência (`timer_set_frequency()`) e os subscribes e unsubscribes do timer (`timer_subscribe_int()`, e o `timer_unsubscribe_int()`) etc...

3.11 Peso relativo dos módulos

Módulo	Percentagem
graphics	15
8042	5
8254	5
keyboard	10
menu	10
mouse	10
movement	15
RTC	10
sprite	15
timer	5

4 Function call graph



5 Detalhes da Implementação

5.1 Object Oriented Language

Tentamos ao máximo abordar o nosso projeto tendo em conta a linguagem orientada a objectos, daí termos criado Enums e Structs para representar maior parte dos objetos e pinguim.

5.2 Best Times e Files

Apesar de files nunca ter sido abordado nas aulas teóricas, decidimos introduzi-lo no nosso projeto, não só porque, como se assemelha a um jogo de arcade, precisar de guardar os "High Scores", porque também já tínhamos decidido implementar isso.

Assim, é possível após jogar o jogo num dia, conseguir verificar os high scores (Best Times), que foram realizados noutro dia. Isto proporciona de certa forma, um desafio ao jogador de tentar ultrapassar sempre o melhor tempo.

Apesar de ter dado algumas dificuldades, tanto na leitura como na escrita para o ficheiro (pois nunca tinha sido abordado), após uma leitura num site na internet, revelou-se mais fácil do que achávamos fazê-lo.

Assim sempre que o programa acaba, o mesmo grava os melhores tempos no ficheiro e data em que foram feitos correspondente e após novo início do jogo, carrega para os arrays de high scores os valores do ficheiro.

5.3 Sprites (XPM)

Maior parte dos sprites utilizados, tanto dos obstáculos como do pinguim foram retirados da internet, sendo que nós tivemos que os modificar de forma a tornar as imagens no tipo .xpm para conseguirmos utilizar no projeto.

Para isto, e para conseguirmos redimensionar os obstáculos e pinguim utilizamos a aplicação GIMP, que nos ajudou bastante nisto.

Todas as letras e números utilizados foram criados no word e posteriormente tratados também pelo GIMP.

5.4 Colisões

No nosso projeto, as colisões têm uma importância extrema, pois sem elas não faria sentido nenhum jogar, pois o desafio seria nulo.

Assim, decidimos tratar das colisões logo num estágio bastante inicialdo projeto

As colisões funcionam usando "hitboxes", ou seja cada obstáculo e pinguim é representado na verdade por um "retângulo", que após embate entre pinguim e obstáculo provoca a morte do pinguim.

Assim criamos uma função que compara as coordenadas de cada uma das hitboxes dos objetos e pinguim e indica se houve colisão ou não, distinguindo também entre colisões permitidas ou não (por exemplo o pinguim poder estar em cima do tronco).

5.5 Layering

Tivemos de certa forma de dividir o nosso projeto em várias camadas, não só para se tornar mais fácil para nós de o organizar como também para o tornar menos confuso.

5.6 Uso dos frames

O jogo funciona com cada interrupt dos periféricos que acontece.

Assim a cada interrupt do timer as posições dos obstáculos e pinguim são atualizadas e é verificado se existiu colisão ou não, ou se houve vitória.

Em caso de vitória, o total de interrupts do timer durante o jogo será guardado, sendo convertido para horas, minutos e segundo, tempo este que irá ser guardado, caso as condições se verifiquem, no array de tempos recorde.

5.7 Doxygen

Toda a documentação do projeto foi realizado recorrendo ao uso do Doxygen, assim como a criação dos gráficos que resumem o projeto.

6 Conclusão

Este projeto, apesar de ter sido muito trabalhoso e nos ter tirado bastantes horas de sono, foi de certa forma muito satisfatório conseguirmos ver o nosso progresso sempre que criávamos uma função nova e víamos o nosso jogo a avançar.

Encontramos algumas dificuldades no início, pois não sabíamos por onde começar, mas após algum estudo sobre o assunto tornou-se bastante mais simples e satisfatório a realização do projeto.

Porém, achamos alguns aspetos negativos no uso do MINIX pois muitas vezes passávamos horas á volta de erros, que acabavam por ser coisas bastante simples mas específicas do MINIX.