U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

João Coelho, up202004846@fe.up.pt

João Mota, up202108677@fe.up.pt

Pedro Lima, up202108806@fe.up.pt

Pedro Landolt, up202103337@fe.up.pt

Turma 5, Grupo 1

Laboratório de Computadores

Licenciatura em Engenharia Informática e Computação

Faculdade de Engenharia da Universidade do Porto

2022 / 2023

Conteúdo

Introdução

O presente relatório descreve o projeto desenvolvido no âmbito do Laboratório de Computadores (LCOM), cujo objetivo foi a criação de um jogo em linguagem C denominado "Defend the Base". Neste projeto, foram implementadas diversas funcionalidades, destacando-se a programação do Timer, do Teclado, do Rato, da Placa de Vídeo e do RTC (Real Time Clock).

O jogo "Defend the Base" consiste em impedir que os inimigos atinjam a base, localizada no canto inferior direito do ecrã, posicionando estrategicamente torres ao longo do caminho. O jogador inicia com 100 moedas e pode arrastar as torres para as posições desejadas utilizando o rato, fixando-as posteriormente com o teclado, de acordo com as instruções exibidas no ecrã. Durante as rondas, os inimigos movimentam-se e as torres disparam. O jogador dispõe de 3 vidas, perdendo uma cada vez que um inimigo alcança a base. O jogo possui um total de 9 rondas, e o jogador vence se conseguir preservar as suas vidas. O jogo foi inspirado no "Flash Element TD", um tower defense onde é necessário adquirir torres para defender a base dos inimigos.

O projeto envolveu a criação de sprites e animações, que foram elaborados a partir do zero no Photoshop guardados em PBM e posteriormente convertidos em ficheiros XPM. Foram desenvolvidas também mecânicas como as balas e o modo noturno, que requereram criatividade e conhecimentos matemáticos para trabalhar com versores e manipulação de cores RGB.

No caso do RTC, foi realizada a sua implementação sem recorrer a interrupções. Inicialmente, tentou-se obter atualizações através de interrupções, mas, devido a dificuldades técnicas, optou-se por descobrir o momento de atualização através de registos.

Este relatório tem como objetivo descrever detalhadamente todas as etapas e implementações realizadas durante o desenvolvimento do projeto "Defend the Base".

Instruções de utilização

O jogador entra no menu, e tem a hipótese de entrar no jogo ou sair.

Figura 1- Main Menu

Ao entrar, no jogo, o jogador tem de impedir que os inimigos (canto superior esquerdo) cheguem à base (canto inferior direito), colocando estrategicamente torres à volta do caminho.

O jogador começa com 100 moedas.

As torres são arrastadas para a posição desejada com o rato e fixadas com o teclado de acordo com as instruções no ecrã. Ao começar a ronda, os inimigos começam a movimentar-se e as torres a disparar. Cada inimigo pode ser acertado por 3 tiros, e o jogador tem 3 vidas, perdendo uma cada vez que um inimigo chega à base. No fim, o jogador vence se não esgotar todas as suas vidas. O jogo tem 9 rondas. No fim de cada ronda, o jogador mantém as suas vidas e ganha 150 moedas.

Figura 2 - Game Screen

Figura 3 - Night Mode

Quando o jogador perde, encontra o ecrã de Game Over, e volta ao modo de texto do Minix.

Figura 4 - Game Over Screen

Estado do Projeto

Funcionalidades	**Dispositivos**	**Estado de Implementação**
Navegação entre menus	Rato e Placa de Vídeo	Completo
Arrastar de Torres	Rato e Placa de Vídeo	Completo
Disparos	Placa de Vídeo	Completo
Upgrade e fixação de torres	Teclado e Placa de Vídeo	Completo
Mostrar a vida e o dinheiro do jogador	Placa de Vídeo	Completo
Escurecimento do ecrã de acordo com as horas	RTC e Placa de Vídeo	Completo
Display das horas	RTC e Placa de Vídeo	Completo

Tabela 1 - Funcionalidades e Estados de Implementação

Dispositivos	**Funcionalidades**	**Interrupções**
Timer	Limita os FPS	Sim
Teclado	Permite dar upgrade a torres e fixá-las no lugar	Sim
Rato	Permite navegar o menu arrastar as torres para o local desejado	Sim
Placa de Vídeo	Mostra todos os elementos relacionados com o jogo (menu, nível com sprites)	Não
RTC	Escurece o ecrã de acordo com as horas reais	Não

Tabela 2 - Dipositivos e Interrupções

Dispositivos

Timer:

A implementação do timer foi feita no ficheiro ***timer.c.***

O timer é usado para limitar a frame rate da placa de vídeo a 60 FPS – função ***is_timer_0_interrupt***, e para calcular o intervalo de tempo entre a saída de inimigos e o intervalo de recarga das torres – funções ***get_time_counter***, ***is_time_interval_elapsed_seconds*** e ***is_time_interval_elapsed_milliseconds***.

Teclado:

A implementação do teclado foi feita no ficheiro ***kbd.c***. O processamento dos inputs é feito através da leitura de scancodes.

O teclado permite a fixação das torres no sítio, através da tecla 'p'. Se a torre for de nível 1, custa 50 moedas, 100 se for de nível 2 e 150 se for de nível 3. A função que permite verificar se podemos pousá-las é ***verifyDrag***. O upgrade das torres é feito na tecla 'u', com o custo de 50 moedas, com a função ***verifyUpgrade***. Para começar uma nova ronda é usado o 'espaço'. Para voltar ao menu anterior é 'ESC'.

Não é utilizado para escrever.

Rato:

A implementação do rato foi feita no ficheiro ***mouse.c***. Ao capturar o rato no MINIX e transformar as suas posições em coordenadas cartesianas, conseguimos ver claramente onde o cursor se situa, e, através do botão esquerdo, selecionar a ação que corresponde à parte do ecrã onde se encontra. O rato permite a navegação no menu – verifica que dentro de coordenadas específicas, a função ***Is_LB_pressed*** é chamada.

No jogo, o rato permite arrastar as torres para a posição pretendida – função ***verifyDrag***.

Para posicionar e dar upgrade às torres, é preciso estar com o cursor em cima delas - funções ***verifyDrag e verifyUpgrade,*** respetivamente.

Placa de Vídeo:

A implementação da placa de vídeo foi feita no ficheiro ***gpu.c.***

A placa de vídeo permite mostrar todas as componentes do jogo, desde o menu até ao nível em si.

Para designs de mapa, inimigos, menus e torres, foram feitos PBMs em Photoshop e convertidos em XPMs, mostrados no ecrã com a função ***draw_xpm*** . A resolução utilizada é 800x600 (VBE 0x115) e as cores de 3 bytes.

Utilizamos double buffering, via copy, para evitar flickering do ecrã. O buffer tem o mesmo tamanho do ecrã. Primeiro, pintamos o buffer (escrevemos as cores de cada pixel no buffer) e depois passamos para a VRAM tudo de uma vez só.

Relativamente a deteção de colisões, os inimigos perdem vida quando a bala lhes toca e as torres não podem ser postas nem no caminho, nem em cima de outra torre – a primeira funcionalidade, através de um if statement que usa um bool gerado na função ***drawBullet***, a segunda através da função ***verifyDrag.***

Todas as fontes utilizadas para dar display a números e letras foram feitas por nós, da mesma maneira que os elementos referidos em cima, e mostradas com a função ***draw_xpm***.

Usamos sprites. Eles são desenhados através da função ***draw_xpm***, assim como os elementos não animados, mas a função é chamada num ciclo, o que permite o desenho contínuo do mesmo elemento em posições diferentes, funcionando como uma animação.

As funções VBE usadas foram ***vg_init, vbe_get_mode_info, vg_exit,sys_privctl, vm_map_phys, sys_int86.***

RTC:

A implementação do RTC foi feita no ficheiro ***rtc.c***.

O RTC permite ler a data e as horas atuais (funções ***get_hours*** e ***get_minutes***.) e controlar o quão escuro o ecrã fica, dessa maneira

As horas são mostradas no menu principal.

Organização do código

Módulo Framework_essencials – **20 %**

Este módulo contém todas funções que correspondem aos essenciais do jogo, como o que fazer quando uma tecla/botão é pressionada ou a função para correr e sair do jogo. Para além disso este módulo gere todas as interrupções necessárias para fazer o jogo funcionar.

Relativamente a estruturas de dados, contém as structs ***position, tower e enemy***, que definem as posições no ecrã, as torres e os inimigos.

Contribuidores:

- Pedro de Almeida Lima

Módulo GPU – **20 %**

Este módulo contém as funções relativamente à placa de vídeo, que permitem fazer a sua configuração e desenhar os XPMs. Algumas foram reutilizadas do Lab 5.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo KBD – **6 %**

Este módulo contém as funções relativamente ao teclado, que permitem lidar com as suas interrupções e ler os scancodes das suas teclas. Algumas foram reutilizadas do Lab 3.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo Mouse – **6 %**

Este módulo contém as funções relativamente ao rato, que permitem lidar com as suas interrupções. Algumas foram reutilizadas do Lab 4.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo Timer – **3 %**

Este módulo contém as funções relativamente ao timer, que permitem lidar com as suas interrupções. Algumas foram reutilizadas do Lab 2.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo RTC – **7 %**

Este módulo contém as funções relativamente ao RTC, que permitem a sua utilização (ler data e horas).

Contribuidores:

- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo Utils – **1 %**

Este módulo contém funções utilitárias que foram fornecidas no Lab 2, e que são utilizadas ao longo do programa.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo Draw – **7 %**

Este módulo contém a função que permite carregar os xpms e desenhar os elementos do jogo.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo Bullet – **3 %**

Este módulo contém a função que permite desenhar as balas que matam os inimigos durante o decorrer do jogo.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo Drag – **3 %**

Este módulo contém as funções que permitem verificar se uma torre está a ser arrastada/a receber um upgrade.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo Game – **20 %**

Este módulo contém funções de preparação e cleanup. Além disso, este módulo contém a função game que é chamada 60 vezes por segundo, sendo a partir daqui que todas as funções responsáveis pelas mecânicas internas do jogo são chamadas.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo Proj – **1 %**

Este módulo contém a função main e invoca as funções definidas em game.c e framework_essencials.c.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Módulo NightMode – **3 %**

Este módulo contém a implementação do “modo noturno”.

Contribuidores:

- João António Teixeira Coelho
- João Guimarães Mota
- Pedro de Almeida Lima
- Pedro Jorge Mendes Jesus Landolt

Grafo de Chamada de Funções

Disponível para consulta em:
p***roj/doc/doxy/html/proj_8c_a2a16f651eccbd248e1ad3b3b924b143b_cgraph.png***

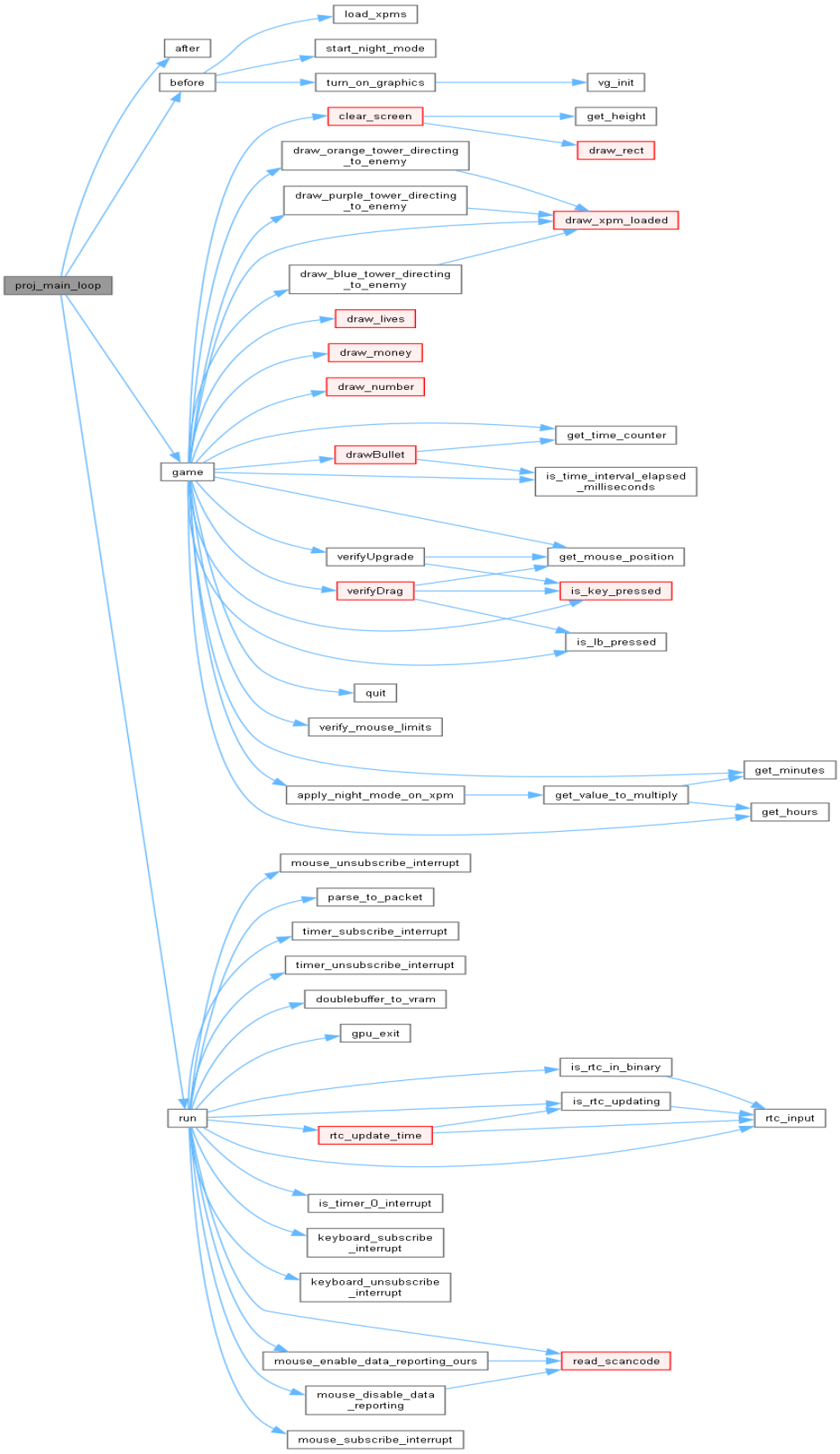

Figura 5 - Grafo de Chamadas de Funções

Detalhes de Implementação

Implementação de mecânicas como as balas e o modo noturno exigiram alguma criatividade e recurso a conhecimentos matemáticos, como versores e trabalhar com RGB.

Implementamos o RTC sem recurso a interrupções. Tentamos, inicialmente, saber quando ele estava a atualizar através de interrupções (UIE - update interrupts), mas, devido a dificuldades técnicas, abandonamos essa estratégia e decidimos descobrir quando o RTC estava a atualizar através de registos.

Conclusões

Conseguimos implementar todas as funcionalidades a que nos propusemos.

Escolhemos este jogo por ser um jogo que permitisse utilizar um grande número de dispositivos, alguns em mais do que uma maneira (exemplo do rato, ao movimentar e clicar).

As tarefas foram distribuídas de modo consensual pelo grupo, tendo todos os elementos participado na elaboração de todos os elementos da entrega final (relatório, documentação e código).

Aprendemos que devido à natureza daquilo que era necessário implementar, foi necessário trabalhar mais do que o previsto no início. Mesmo assim, conseguimos atingir o objetivo final.

Concluímos então que o projeto foi bem-sucedido e importante para o nosso crescimento no que toca a lidar com dispositivos I/O.