

Ano Letivo 2017/18

Laboratório de Computadores

SHOOT MONSTERS

Grupo T8G07: Nuno Tiago Lopes up201605337@fe.up.pt

Diogo Lousa Ribeiro up201605324@fe.up.pt

Índice

1. Instruções de utilização	3
1.1. Menu inicial	3
1.2. Shoot Monsters.....	4
1.3. Highscores.....	5
1.4. Game Over	5
2. Estado do projeto	7
3. Organização/Estrutura do código	9
3.1. Shoot Monsters (game.h)	9
3.2. Pixmap.....	9
3.3. Sprite	9
3.4. Highscore	9
3.5. Keyboard.....	10
3.6. Mouse.....	10
3.7. Timer.....	10
3.8. RTC	10
3.9. Vídeo_gr.....	10
4. Gráfico de chamada de funções.....	11
5. Detalhes de implementação	12
5.1. State Machine.....	12
5.2. Utilização de pixmaps	12
5.3. Implementação do RTC	13
5.4. Monstro	13
6. Avaliação da Unidade Curricular	14
7. Instruções de instalação	15

1. Instruções de utilização

1.1. Menu inicial

Ao iniciar o jogo, surge o menu inicial. (Figura 1).



Figura 1 - Menu Inicial

Este permite ao utilizador escolher entre iniciar o jogo propriamente dito, ver os highscores do jogo ou sair do mesmo.

Para navegar no menu inicial, o utilizador tem de usar as teclas "UP" e "DOWN" do teclado e seleccionar a opção que deseja, utilizando a tecla "ENTER". A opção seleccionada está destacada com uma cor verde que a diferencia das restantes opções.

No caso de querer sair do jogo, o utilizador pode seleccionar a opção "QUIT" ou então premir a tecla "ESC".

1.2. Shoot Monsters

Quando a opção “PLAY” é selecionada, aparece uma barra verde no topo da tela de jogo com informação sobre o número de balas disponíveis (BULLETS), o número de vidas (LIVES), tempo de jogo (TIME) e a pontuação atual (SCORE). De imediato começa, também, a aparecer e a desaparecer o monstro do jogo.

O objetivo do jogo é matar o monstro o maior número de vezes possíveis, utilizando uma mira controlada pelo periférico rato. O movimento do rato desloca a mira e o botão esquerdo do mesmo faz a arma disparar. Cada vez que o utilizador acerta no monstro e o mata, ganha 10 pontos que são automaticamente adicionados à pontuação.

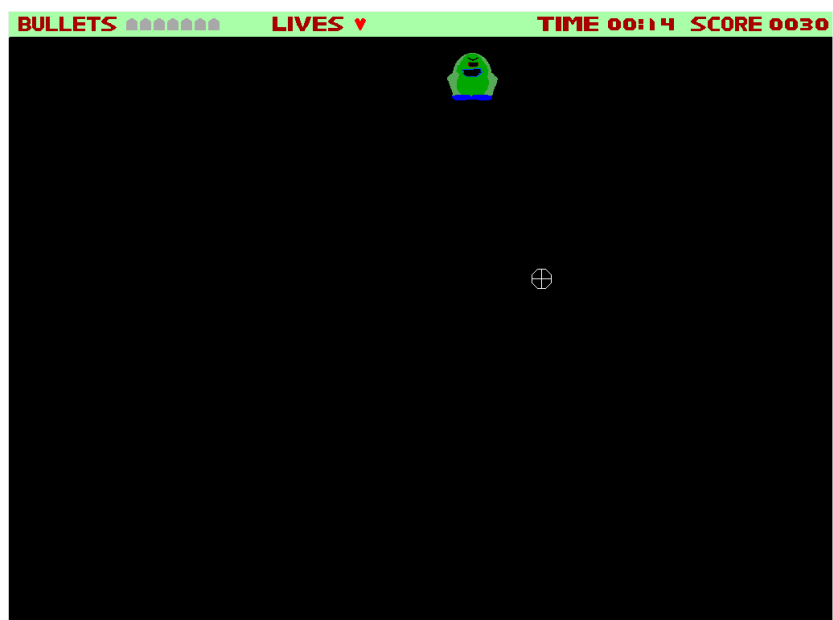


Figura 2 - Tela de jogo

A arma tem um limite de 10 balas que podem ser recarregadas pressionando a tecla “ESPAÇO” do teclado.

O jogo inicia-se com 5 vidas. O utilizador perde quando esgota as suas vidas. Uma vida é perdida quando o player não acerta num monstro, acontece normalmente quando já foram esgotadas as 10 balas e se tenta disparar sobre o monstro sem recarregar as mesmas ou então quando o monstro desaparece muito rápido e o utilizador não lhe consegue acertar antes deste desaparecer.

A dificuldade do jogo vai aumentando com o tempo, pois o monstro fica menos tempo visível no ecrã do jogo.

1.3. Highscores

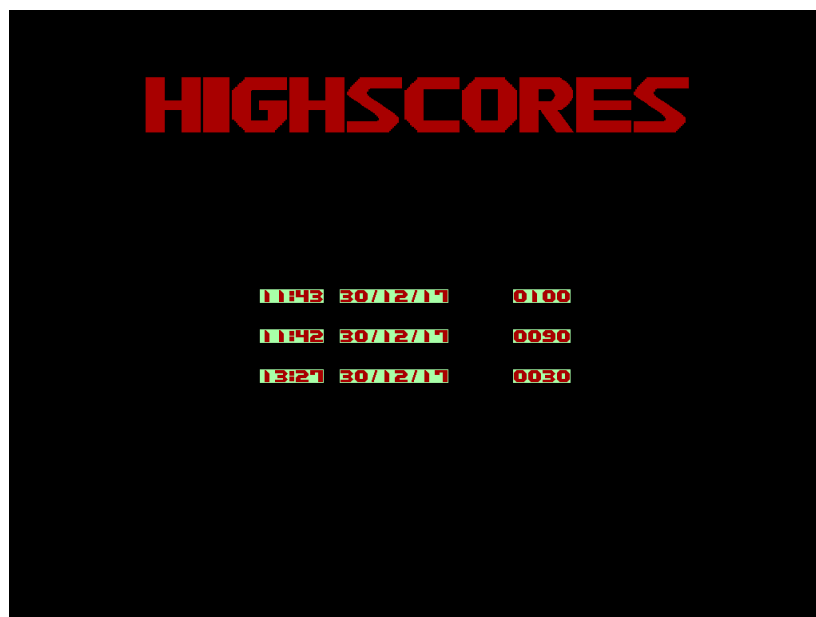


Figura 3 - Highscores

Nesta opção o player pode visualizar as 3 melhores pontuações realizados, podendo ver o dia, mês e ano em que foram conseguidos, bem como a hora e os minutos.

Para retornar ao menu inicial, o utilizador só necessita de carregar na tecla “ESC” do teclado.

1.4. Game Over

Quando o utilizador perde o jogo surge uma mensagem a dizer “GAME OVER” como podemos ver na figura 4. Na mesma imagem podemos, também, visualizar o tempo de jogo e o score obtido nessa jogada.

Para voltar ao menu inicial o player tem de pressionar a tecla “ESC” do teclado.



Figura 4 - Game Over

2. Estado do projeto

Dispositivo	Função	Interrupções?
Timer 0	Frame Rate e cronómetro.	Sim
Teclado	Navegação no menu, recarregar arma e sair do jogo.	Sim
Rato	Movimento da mira, disparar a arma.	Sim
RTC	Obter data e tempo atual para guardar com os highscores.	Não
Placa Gráfica	Apresentação visual dos aspetos gráficos do jogo (menu, monstro, mira, etc,...).	Não

Mais concretamente, os dispositivos foram utilizados do seguinte modo:

- Timer 0: as interrupções do timer 0 permitem controlar o tempo entre cada alteração da posição do monstro e também um cronómetro de jogo. Estas interrupções são processadas no ficheiro game.c na função timer_handler.
- Teclado: é utilizado para navegar entre as diferentes opções do menu e seleccioná-las. Durante o jogo é serve para fazer o recarregamento da arma (tecla “espaço”) e para sair do mesmo (tecla “ESC”). O teclado é processado na função keyboard_handler do ficheiro game.c.
- Rato: é o periférico principal do jogo pois é utilizado para mover a mira e para a disparar (botão esquerdo). É interpretado na função mouse_handler do ficheiro game.c.

- RTC: é utilizado em modo de polling para obter a data (dia, mês e ano) e o tempo (horas e minutos) de quando foi obtida uma das 3 pontuações máximas. Pode ser encontrado nas funções `rtc_get_time` e `rtc_get_date` do ficheiro `rtc.c`.
- Placa gráfica: as suas funções podem ser encontradas nos ficheiros `video_gr.c` e `sprite.c`.

3. Organização/Estrutura do código

O projeto foi realizado com a intenção de ficar bem estruturado para alguém que não tenha realizado o trabalho o consiga perceber. As características de cada módulo e o peso relativo no projeto são apresentados abaixo.

O elemento Diogo Ribeiro ficou responsável pelo módulo Pixmap e o elemento Nuno Lopes ficou responsável por todos os outros módulos.

3.1. Shoot Monsters (game.h) 30%

É o módulo principal do projeto, responsável por ligar todos os outros. Serve para coordenar todas as operações que resultam de cada um dos módulos. Contém os handlers dos dispositivos teclado, timer e rato e os handler da informação apresentada na barra verde no topo da tela de jogo.

3.2. Pixmap 6%

Contém todas as “imagens” (xpm) utilizadas no jogo.

3.3. Sprite 12%

Este módulo é responsável pela criação de todas as “sprites” (imagens) do jogo. É o módulo acedido por todos os outros quando se pretende mostrar algo na tela de jogo. Contém a struct de um sprite.

3.4. Highscore 10%

Módulo que gere os highscores do jogo e trata em ler e guardar a informação num ficheiro (highscores.txt).

3.5. Keyboard 5%

Responsável pelos subscribe e unsubscribe das interrupções do teclado.

3.6. Mouse 10%

Responsável pelos subscribe e unsubscribe das interrupções do rato e pela alteração da posição da mira na tela de ecrã. Contém a struct do rato.

3.7. Timer 5%

Responsável pelos subscribe e unsubscribe das interrupções do timer.

3.8. RTC 7%

Módulo que obtém a data e o tempo atual do computador.

3.9. Vídeo_gr 15%

Este módulo é responsável por mostrar as “imagens” (xpm), a mira (scope) e limpar a tela do ecrã.

4. Gráfico de chamada de funções

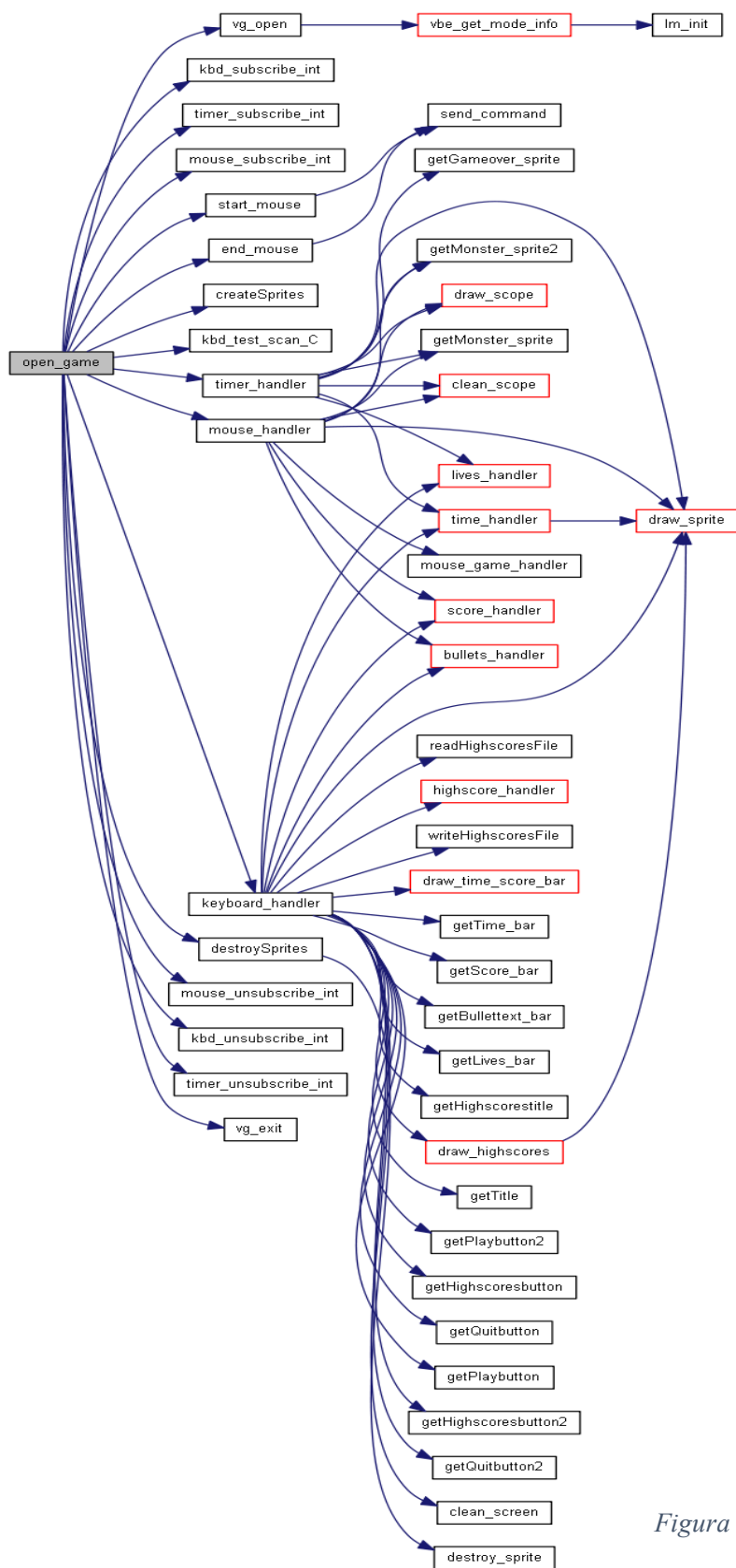


Figura 5 - Gráfico de chamada de funções

A função `open_game` é chamada na função `main` do ficheiro `proj.c`.

5. Detalhes de implementação

O projeto foi realizado tendo por base o que foi desenvolvido nos labs, no entanto foram feitas alterações necessárias para estes poderem ser utilizados no projeto.

5.1. State Machine

Primeiramente tivemos de implementar uma state machine que controla o estado do jogo, isto é, se está no menu inicial e em que opção deste está, se está nos highscores, se o player já está a jogar ou se perdeu o jogo.

Os diferentes dispositivos só são interpretados dependendo da posição da state machine.

Concluindo, a state machine é a base de todo o jogo.

5.2. Utilização de pixmaps

Na realização deste projeto decidimos utilizar apenas xpm em vez de bitmaps uma vez que, a nosso ver, o importante não seria o aspeto visual do jogo, mas sim a sua funcionalidade.

Deparamo-nos com alguns problemas, principalmente porque foi feito include do pixmap.h em vários ficheiros e a chamada muito repetitiva da função read_xpm que, por sua vez, ocupava muita memória do sistema Minix e o impedia de correr o nosso código. A solução para este problema foi a criação de dois ficheiros (sprite.h e sprite.c) que contêm todos os xpm lidos, e colocados numa struct, uma única vez, ou seja, para cada um só é chamada exclusivamente uma vez a função read_xpm. Estas structs são acedidas em qualquer lado do projeto.

5.3. Implementação do RTC

A implementação do dispositivo RTC não surtiu grandes complicações pois é unicamente utilizado para ler a data (dia, mês e ano) e o tempo (horas e minutos) do momento em que é atingida uma pontuação máxima que, por sua vez, é guardada num ficheiro (highscores.txt) e é acedida quando no menu inicial é selecionada a opção “HIGHSCORES”.

5.4. Monstro

A partir das interrupções do timer 0, é possível realizar o aparecimento e desaparecimento do monstro do jogo, para tal, foram utilizados dois timers. A cada 5 segundos a velocidade com que o monstro aparece é aumentada, reduzindo assim o tempo que fica na mesma posição da tela de jogo e aumentando a dificuldade do jogo.

A posição onde o monstro aparece é aleatória, porém teve de ser interpretada de modo a que o monstro não ficasse cortado a meio devido aos limites da tela do jogo. O mesmo teve de se fazer para a mira aparecer sempre inteira.

6. Avaliação da Unidade Curricular

A nossa avaliação para esta unidade curricular tem aspetos positivos e negativos.

Positivamente destacamos o facto de termos tido a oportunidade de aumentar os nossos conhecimentos de linguagem C e um pouco de assembly. Também destacar o facto de termos aprendido como funcionam os dispositivos, teclado, rato, timer e o rtc.

Pela negativa sentimo-nos um pouco injustiçados por termos a aula à terça-feira e os limites de entrega dos labs não serem iguais para todos, ou seja, os alunos que têm a aula prática no final da semana têm mais tempo para realizar as atividades.

Por fim, e ainda falando dos aspetos negativos, o trabalho solicitado nesta unidade curricular é, comparativamente às restantes unidades curriculares, muito mais exigente.

7. Instruções de instalação

O ficheiro contido na pasta “conf” tem de ser copiado com permissões de root para a pasta /etc/system.conf.d, pode ser feito através do comando “cp path-to-the-folder/conf/proj /etc/system.conf.d”. Também é necessário verificar em que pasta está o ficheiro “highscores.txt” e alterar se necessário nas funções writeHighscoreFile e readHighscoreFile que estão no ficheiro highscores.c