Ano Letivo 2017/2018

FUGIX

Relatório

Laboratório de Computadores

LCOM1718-T7G01

João Alves – up201605236
Amadeu Pereira - up201605646

Índice

1. Introdução

Para a unidade curricular de Laboratório de Computadores foi-nos pedido a criação de um projeto em que fossem utilizados os periféricos com que tinhamos trabalhado nas aulas laboratoriais desta mesma cadeira.

Assim, lembramo-nos de fazer um jogo em que o objetivo seria esquivar de vários objetos, de forma a sobreviver o maior tempo possível, tendo como objetivo conseguir o melhor score possível. Foi assim que surgiu o FUGIX.

2. Instruções de utilização

2.1 Menu Principal

O jogador pode interagir com o menu principal utilizando o rato (selecionando uma das opções dadas, de forma a escolher o procedimento do jogo). No entanto, também é possível começar um jogo utilizando a tecla ENTER, e sair utilizando o ESC.

2.2 Dica

Ao começar um jogo novo, o jogador vai deparar-se com um menu que lhe indica uma forma de conseguir atingir uma pontuação mais elevada, esta pode ser passada à frente utilizando a tecla ENTER.

2.3 Highscores

Se for escolhida a opção de highscores no menu principal vão ser mostradas, ao jogador, o top 3 de melhores scores atingidos neste jogo, assim como o dia e a hora em que foram obtidos, sendo possível sair através das teclas ENTER e ESC.

2.4 Jogo

Quando iniciado o jogo, é dada, ao jogador, uma nave que ele pode controlar através do rato. O objetivo deste é não ser atingido pelos asteroides que estão a cair de cima, estes têm vários tamanhos (os grandes têm 3 vidas, os médios, 2 vidas e os pequenos apenas 1) e várias velocidades. De maneira a ajudar o jogador foram implementados tiros, assessiveis através do botão esquerdo do teclado e um escudo que o torna invulneravel durante algum tempo (este pode ser ativado com a tecla Z, tendo, no entanto, um tempo minimo de espera até ser possível ativar outra vez). Ao longo do jogo a dificuldade deste aumentará.

2.5 Gameover

Sendo este um jogo de sobrevivência infinito, nunca haverá um final. Assim, após a perda do jogo aparecerá o menu de gameover, sendo dadas várias opções ao jogador para decidir o que fazer a seguir. Se o score atingido for melhor que o top 3 anterior este ocupará a devida posição no pódio.

3. Estado do projeto

Periférico	Resumo da função	Interrupção?
TIMER	Temporização do jogo (atualizando o score) , atualização de estados e frame rate.	SIM
TECLADO	Saída do jogo , passagem do gameover para o menu ou para voltar a jogar, e passagem da tip para o jogo em si.	SIM
RATO	Navegação no menu e escolha da opção , controlo da nave em jogo.	SIM
PLACA GRÁFICA	Apresentação visual dos gráficos do jogo (menu , constiuintes de jogo ,etc)	NÃO
RTC	Forma de guardar os highscores (utilizando a data e a hora como identificadores do score)	NÃO

3.1 Timer

As interrupções do timer permitem fazer o update e controlo do estado do jogo chamando a funcção timer_handler() (proj.c).
Também serve para atualizar a imagem do ecrã.
É também utilizado um contador para incrementar o score do jogo.

3.2 Teclado

O teclado é usado para a fazer a passagem de certos estados. Para tal é chamada a função keyboard_handler (proj.c) a cada interrupt do mesmo. Permite recomeçar o jogo quando se encontra no estado de gameover (pressionando o enter) ou passar para o menu (utilizando o esc).

3.3 Rato

O rato é utilizado para navegar pelo menu e para seleccionar a opção pretendida. É também utilizado para controlar a nave em jogo.

Tudo isto é feito com a chamada da função mouse_handler (proj.c) a cada interrupt.

3.4 Placa Gráfica

A placa gráfica é utilizada no modo 0x114 com uma resolução 800*600 , utilizando uma palete de 16bits de RGB.
Foi criado um double_buffer para diminuir o flickering no movimento de imagens (.bmp) na atualização de frame.
De um modo geral a placa gráfica é utilizada para a visualização dos gráficos do jogo.

3.5 RTC

O rtc é utilizado de forma a poder gravar um score (quando este entra para o top de highscores). O score fica gravado identificado com a hora e a data em que foi obtido.

4. Organização/Estrutura do código

4.1 bitmap.c/bitmap.h

Este módulo é utilizado para a interação com imagens (.bmp) , é da autoria do aluno Henrique Ferrolho e o código pode ser encontrado aqui: http://difusal.blogspot.pt/2014/09/minixtutorial-8-loading-bmp-images.html

No entanto foram feitas algumas alterações para suportar colisões.
Membro do grupo responsável : Amadeu Pereira

4.2 collisions.c/collisions.h

Este módulo foi criado como método para trabalhar à volta das colisões entre nave-asteroide ou asteroide-bala , foi criada uma mask para simular a ocupação de cada objecto na placa gráfica e assim facilitar a identificação de uma colisão.

Membro de grupo responsável : Amadeu Pereira

4.3 highscores.c/highscores.h

Módulo criado para trabalhar com um ficheiro de texto. Envolve a criação de uma struct (score_t) e a definição global de 3 variáveis(do tipo score_t) correspondentes aos 3 melhores scores.
Foram criadas funções que trabalham com o ficheiro de modo a poder ler e escrever do mesmo , “atualizando” os valores das variáveis globais.

Membro de grupo responsável : João Alves

4.4 i8042 / i8254

Apenas para definição de macros.

Membro responsável : João Alves

4.5 input.c / input.h

Neste módulo foram criadas structs e respectivas funções que têm como objetivo facilitar a “leitura” de um input do rato ou do teclado de modo a que os estados e as suas mudanças sejam feitas de acordo com os mebros das structs.

Membro responsável : Amadeu Pereira

4.6 keyboard.c / keyboard.h

Funções já utilizadas no lab respetivo ao teclado , que permitem trabalhar com o mesmo e suas interrupções. De um modo geral , fazer subscribe, unsubscribe, ler o scancode , e escrever/ler comandos.

Membros responsáveis : João Alves e Amadeu Pereira

4.7 mouse.c / mouse.h

Módulo onde são implementadas as funções que trabalham com o rato. Foram assim criadas as funções para subscribe, unsubscribe , ler/escrever comandos , para ler do outbuf , para ver o status e funções para alterar certos modos do rato.

Membros responsáveis : João Alves e Amadeu Pereira

4.8 objects.c / objects.h

Destacamos a importância deste módulo , pois foi fundamental para o desenvolvimento do projeto.
É aqui que são implementadas as funções que criam e destroem todos os “objetos” criados e utilizados em jogo , desde asteroid , bullet ,spaceship ,

background , etc. Todas estas structs vão ser inicializadas numa variável global gamePtr (Game_t) utilizando as funções aqui desenvolvidas.
No fim do jogo são também utilizadas as funções que permitem dar free dos apontadores de cada struct.

Membros responsáveis : João Alves e Amadeu Pereira

4.9 proj.c / proj.h

Módulo principal onde é iniciado o jogo e a variável global correspondente. Aqui é criada a função com o ciclo gobal de interrupções onde são chamadas as funções : mouse_handler, keyboard_handler, timer_handler (fundamentais). Nestas funções é feito o handling de eventos e estados do jogo , das atualizações e mudanças dos gráficos do mesmo.

Membros responsáveis : Amadeu Pereira e João Alves

4.10 rtc.c / rtc .h

Módulo onde são implementadas as funções para obter a data e a hora do dia.

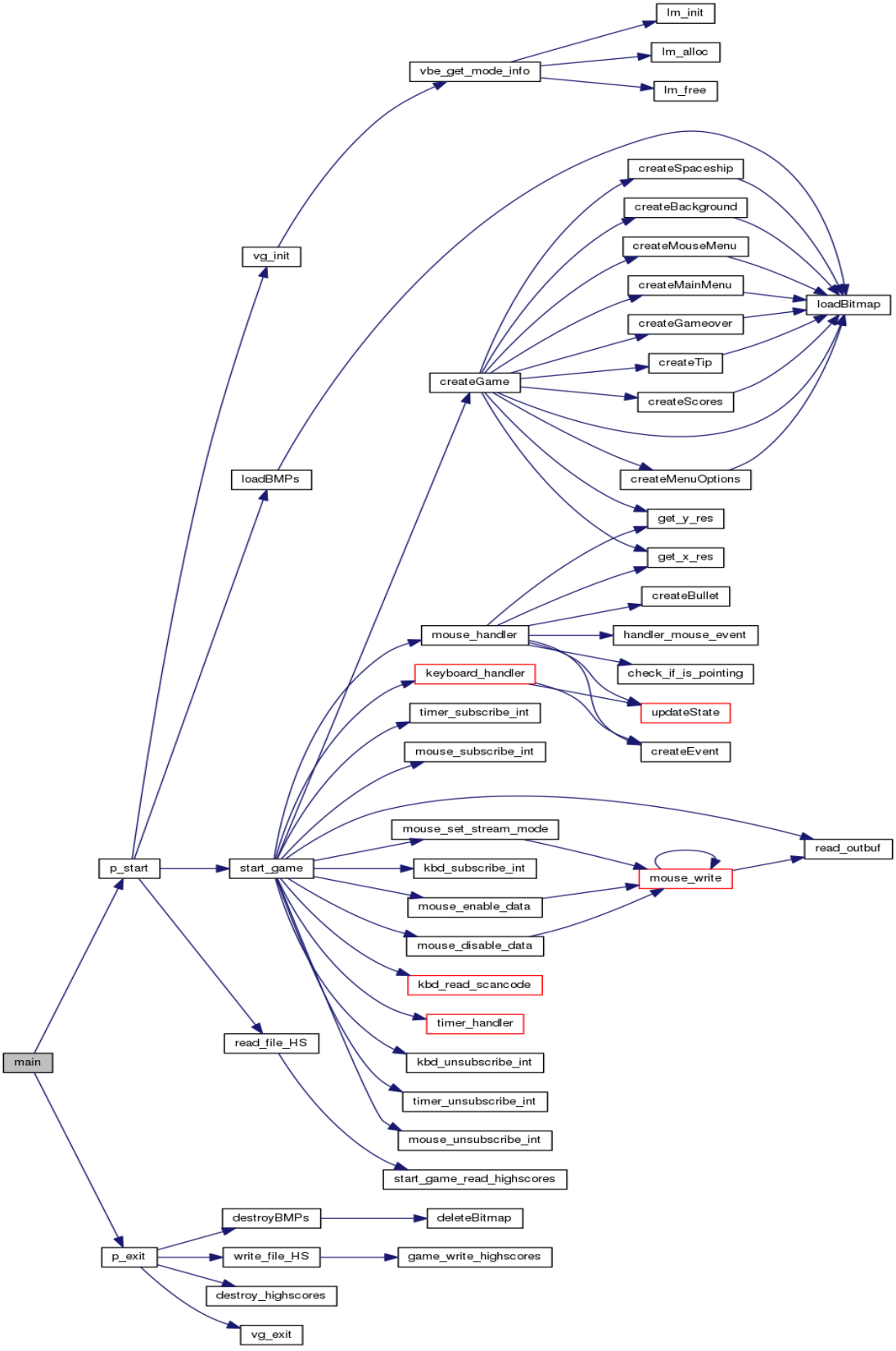
Membro responsável : João Alves

4.11 timer.c / timer.h

Neste módulo foram implementadas as funções que manipulam o timer como subscribe e o unsubscribe e uma função para fazer uma “espera” de modo a atrasar as restantes instruções.

Membros responsáveis : João Alves e Amadeu Pereira

5. Gráficos de chamada de funções




start_game
timer_subscribe_int
mouse_subscribe_int
kbd_subscribe_int
mouse_set_stream_mode
mouse_enable_data
mouse_disable_data
kbd_read_scancode
kbd_status
keyboard_handler
handler_kbd_event
timer_handler
drawGame
updateGame
drawGameOver
destroyGame
drawHighScores
bufferSwap
drawMenu
drawGameTip
drawMainMenu
drawMenuOptions
drawMouseMenu
kbd_unsubscribe_int
timer_unsubscribe_int
mouse_unsubscribe_int
mouse_write
wait_inbf
wait_outb
read_outbuf
mouse_handler
handler_mouse_event
check_if_is_pointing
createBullet
updateState
createEvent
updateHighScores
resetGame
createGame
createMainMenu
createGameover
createTip
createScores
createMenuOptions
createSpaceship
createBackground
createMouseMenu
loadBitmap
get_x_res
get_y_res

6. Detalhes de implementação

Achamos relevante chamar a atenção de que na implementação das colisões foi quase que criado um ecrã virtual binário do jogo, em que os pixéis correspondentes aos asteróides são colocados a 1 neste ecrã virtual. No entanto, sendo o jogo 800x600 pixéis chegamos à conclusão de que a melhor implentação seria a criação de um array duplo do chars, tal array teria como dimensões 100x600 sendo, assim, utilizados todos os bits deste array, melhorando a eficiência do nosso projeto.

Para detetar as colisões alteramos a função drawBitmap para duas variações. A primeira foi uma que continuava a desenhar o objeto dado mas também colocava as todas as coordenadas ocupadas pelo objeto a 1 na máscara utilizado para as colisões. A segundo desenha na mesma o objeto dado, no entanto, retorna 1 caso tenha ocurrido uma colisão, no sitio onde o objeto que queremos verificar se encontra.

Na criação dos asteróides definimos 3 variáveis que nos iriam ajudar posteriormente (a velocidade minima dos asteróides, a máxima e o tempo de intervalo entra cada “spawn”). Estas variáveis, ao longo do jogo, vão sendo alteradas de forma a ser possível aumentar a dificuldade do jogo (estas alterações fazem com que apareçam asteroides cada vez mais rápido e que estes se movam com mais velocidade).

7. Instalação

O ficheiro “proj” que se encontra na pasta conf deve ser copiado para a pasta /etc/system.conf.d/ utilizando o comando cp (é necessário permições de root).

8. Avaliação global da unidade curricular

Achamos que esta unidade curricular é muito vantajosa para nós, pois permite-nos conhecer como é que cada periférico funciona e o porquê de funcionar assim, melhorando assim o nosso conhecimento enquanto programadores.

No entanto achamos que também existem pontos negativos, tais como: a notória diferença de dificuldades entre as turmas que têm aulas práticas às terças com os que têm às sextas. Parecendo que não, estes vão ser sempre beneficiados uma vez que têm ajuda daqueles que já tiveram que entregar o lab, quase durante uma semana inteira. Também achamos que os pdf’s nem sempre são muito explícitos, às vezes, ao fim de ler e reler os pdf’s e outras fontes de informação continuavámos sem perceber bem o que fazer e como fazer.