

"Duck Hunt – Minix Edition"



Unidade Curricular: Laboratório de Computadores(LCOM)

Ano: 2017/2018

Turma: 3

Grupo: 01

Autores: Fernando Jorge Alves – up201605270

João Carlos Maduro – up201605219

Conteúdo

1. Introdução.....	3
2. Instruções de Utilização	4
2.1. Menu Inicial	4
2.2. GAME A – SINGLEPLAYER	4
2.3. GAME B – HUNTER VS DUCK	5
Duck.....	6
Hunter	6
2.4. Game Win	7
2.5. Game Over	7
2.6. HIGHSCORE	8
3. Módulos Implementados	9
3.1. Timer	9
3.2. Teclado	9
3.3. Rato	9
3.4. Placa Gráfica	10
3.5. RTC	10
3.6. Porta de Série	10
4. Estrutura e Organização do Código.....	11
4.1. bitmap.c	11
4.2. duck.c	11
4.3. extras.c.....	11
4.4. game.c.....	11
4.5. keyboard.c	12
4.6. main.c.....	12
4.7. menu.c	12
4.8. mouse.c.....	12
4.9. points.c.....	12
4.10. rtc.S	13
4.11. serial_port.c	13
4.12. timer.c	13
4.13. vbe.c e video_gr.c.....	13
5. Detalhes de Implementação	15
5.1. Porta de Série	15
6. Conclusão.....	16
7. Apêndice	17

1. Introdução

Para a unidade curricular Laboratório de Computadores foi-nos proposto realizar um projeto em que pudéssemos utilizar os vários módulos lecionados, mas agora em conjunto. Sendo nós ávidos jogadores de jogos de vídeo, decidimos recriar um jogo popular recorrendo aos conhecimentos que adquirimos nas aulas de LCOM e chegamos à conclusão que o jogo que melhor se enquadra nesse perfil seria o muito popular “Duck Hunt”, e assim decidimos recriar o mesmo para o sistema operativo MINIX, nascendo assim o “Duck Hunt – Minix Edition”.

O Duck Hunt original foi desenvolvido pela Nintendo, em 1984, e lançado para a consola NES(Nintendo Entertainment System). O objetivo do jogo consistia em matar todos os patos presentes no ecrã, recorrendo apenas a um número limitado de balas. O jogo encontra-se dividido em rondas e por cada ronda o número de patos aumenta, mas o número de balas mantém-se constante. Existe também um modo 2 jogadores, em que o segundo jogador controla o pato que tem de fugir da mira do jogador 1.

2. Instruções de Utilização

2.1. Menu Inicial



Imediatamente após executar o programa, é apresentado ao utilizador o Menu Inicial do jogo. Recorrendo às setas de cima e de baixo do teclado, o utilizador é capaz de navegar e seleccionar uma das 4 opções existentes: GAME A – SINGLEPLAYER, GAME B – HUNTER VS DUCK, HIGHSCORE e EXIT.

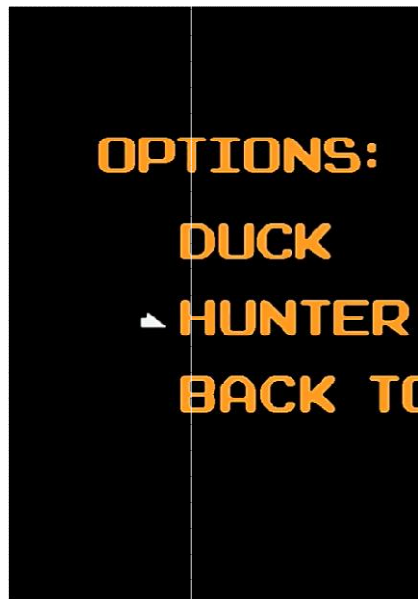
2.2. GAME A – SINGLEPLAYER



No Game A é onde decorre o jogo principal. Consoante o número da ronda, variará o número de patos, que são gerados atrás do arbusto. Após gerados, os patos

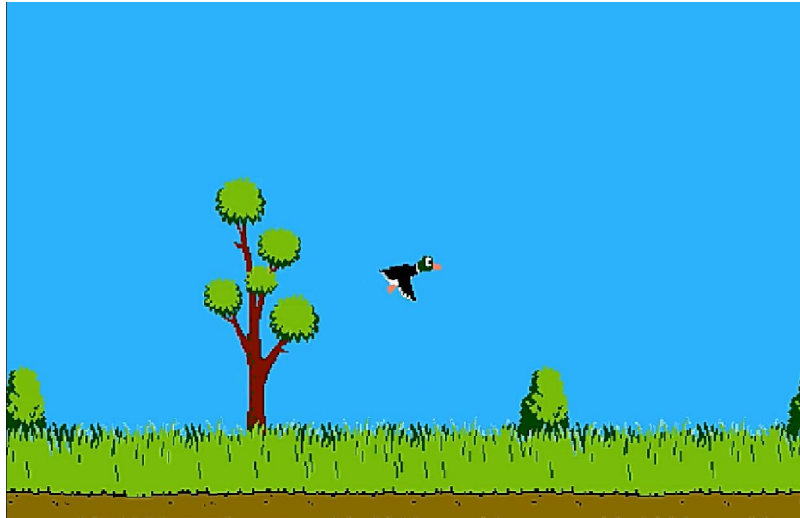
começarão a voar pela tela do jogo e o jogador terá de acertar em todos eles para poder passar à ronda seguinte. Após concluídas as 10 rondas, será apresentado ao jogador o ecrã de vitória. Caso o jogador, em alguma ronda tenha gasto todas as balas e ainda existam patos a voar, ser-lhe-á mostrado o ecrã de derrota.

2.3. GAME B – HUNTER VS DUCK



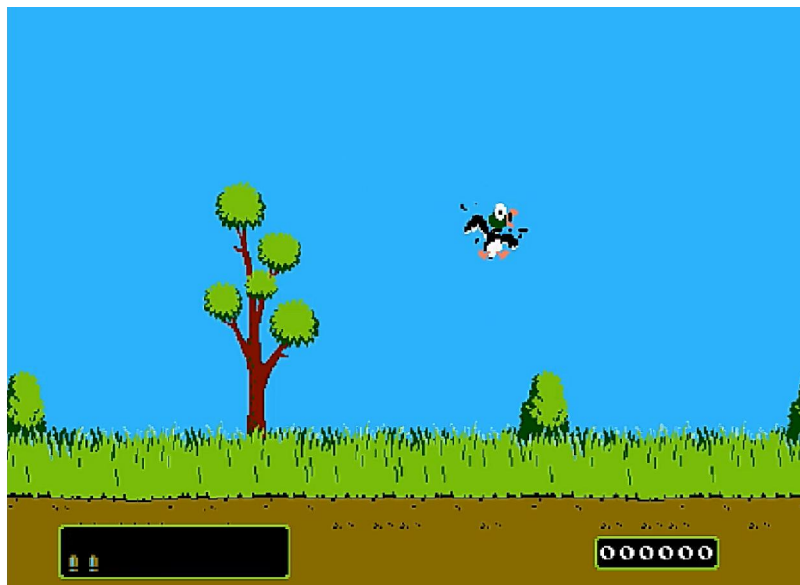
Se o utilizador escolher a opção Game B, será reencaminhado para um menu onde pode escolher entre jogar com o caçador ou com o pato, ou regressar o menu inicial. Para usufruir desta opção é necessário que um dos dois utilizadores(cada um no seu respetivo computador) selecione o pato e o outro o caçador, e que os dois computadores estejam ligados pela porta de série.

Duck



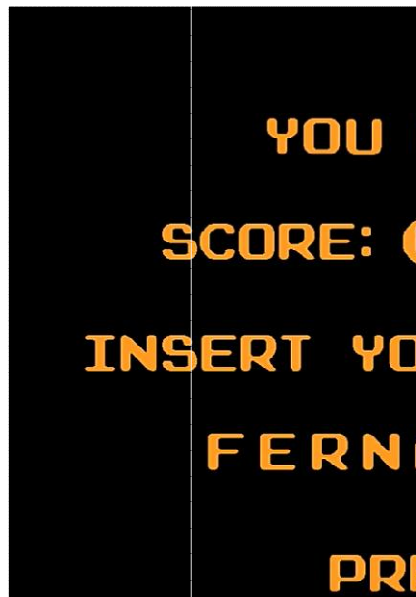
Caso o utilizador tenha escolhido jogar como Pato, poderá controlar o voo de um único pato no ecrã recorrendo às teclas W, A, S e D. O objetivo deste modo consiste em evitar ser atingido pelas balas disparadas pelo caçador, ou seja o outro jogador. Mas para tornar a situação mais difícil, o Pato não sabe para onde está a apontar o caçador e como tal, o jogador que controla o pato, não sabe em que posição do ecrã se encontra a mira. Se o pato sobreviver a uma das 10 rondas, ganha o jogo.

Hunter



No caso, de o utilizador optar por jogador pelo caçador, terá como missão acertar no pássaro. Este jogador ganha se conseguir acertar no pato em todas as 10 rondas, sendo que a cada ronda o número de máximo de balas decresce em uma bala.

2.4. Game Win



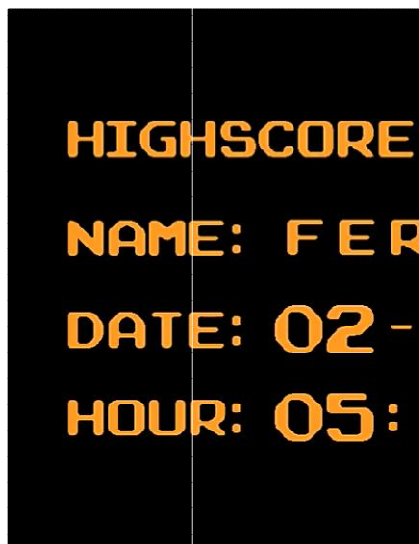
Em caso de vitória, o jogador será deparado com uma tela com o título “You Win”. Se o utilizador jogou o modo Game A, também lhe será apresentado a pontuação que obteve nessa sessão e ainda um espaço para poder escrever o seu nome, para a pontuação ficar associada a esse nome, caso a pontuação tenha batido o recorde anterior.

2.5. Game Over



Caso o jogador tenha perdido, ser-lhe-á mostrado uma tela com título “Game Over”. Tal como no cenário de vitória no Game A, no estado de derrota do mesmo modo será apresentado a pontuação obtida e será possível ao jogador escrever o seu nome, pois mesmo tendo perdido, pode ser que tenha obtido um novo recorde.

2.6. HIGHSCORE



Nesta opção do menu principal, é possível observar a atual pontuação mais alta obtida no modo Game A, a hora em que foi obtida e o nome de quem a obteve. Se não houver pontuação registada, o utilizador será avisado de tal. É possível apagar o recorde atual pressionando a tecla R.

3. Módulos Implementados

No trabalho final, fomos capazes de cumprir todas as metas que estabelecemos na especificação do projeto. Conseguimos implementar todos os módulos e ainda um deles em assembly, sendo este o RTC.

Periférico	Resumo do uso do periférico	Interrupção
Timer	“Refresh” dos gráficos e controlo da lógica do jogo	SIM
Teclado	Movimento do pato e da setinha do menu	SIM
Rato	Deslocamento no menu e interação com as opções	SIM
Placa Gráfica	Representação visual da lógica do jogo	NÃO
RTC	Registo das pontuações e para o relógio do menu	NÃO
Porta de Série	Comunicação e sincronização entre computadores	SIM(receção)

3.1. Timer

Este módulo é utilizado para atualizar os gráficos, 60 vezes por segundo. Também utilizámos o timer para controlar a duração de determinadas ações do jogo. As funções do Timer encontram-se implementadas nos ficheiros: timer.c, timer.h e i8254.h. Temos funções que recorrem a interrupções do timer nos ficheiros: duck.c.

3.2. Teclado

Utilizamos o teclado para navegar e interagir com a interface do jogo, como por exemplo menus, onde as suas interrupções servem para atualizar gráficos, juntamente com o timer, porque o ultimo sozinho não atualiza instantaneamente. No jogo em si é usado para controlar o pato no modo GAME B. Este modulo encontra-se desenvolvido em keyboard.c, keyboard.h e i8042.h. Desenvolvemos também estruturas que se auxiliam do keyboard em extras.c e em menu.c.

3.3. Rato

Utilizamos o rato maioritariamente para controlar a mira do caçador, quer no modo Game A, quer no modo Game B(caso o utilizador esteja a controlar o caçador). Quando pressionado o botão esquerdo do rato é disparada uma bala. Este módulo foi implementado nos ficheiros: mouse.c, mouse.h. E temos estruturas que recorrem ao mouse no ficheiro extras.c.

3.4. Placa Gráfica

Utilizamos a placa gráfica para gerar os gráficos do projeto. O modo que decidimos utilizar foi o 0x117, cuja resolução é 1024x768 com uma paleta de cores de 16 bits (5:6:5).

Para manter a atualização de frames consistente e suave, implementamos double buffer. As imagens utilizadas durante os 2 modos de jogo, tais como os patos ou o fundo, foram retirados do jogo original, atualizando apenas o tamanho para a nossa resolução. O font usado para as letras e os dígitos também não é da nossa autoria, escolhemos um que se adequava ao estilo do jogo.

A implementação deste módulo encontra-se em: `vbe.c`, `vbe.h`, `graphics.c`, `graphics.h`, `video_gr.c`, `video_gr.h`, `bitmap.c` e `bitmap.h`.

3.5. RTC

O RTC foi utilizado para termos acesso aos valores de data/hora, que usámos no menu inicial para mostrar a hora atual, e para guardar a data e hora do highscore, que pode ser mais tarde visto na secção apropriada do menu.

A implementação deste módulo encontra-se em `rtc.S`, `rtc_macros.h`.

3.6. Porta de Série

Utilizámos a porta de série para envio de informação entre o pato e o caçador. O pato envia se desistiu e se ocorreu mudança de movimento. O caçador envia, se desistiu, se conseguiu abater o pato e também para decidir quem foi o vencedor da partida. Em ambos os jogadores(Pato e Caçador) é feita uma interrupção cada vez que se há dados a serem recebidos. Foi também desenvolvida uma função onde os dois jogadores enviam dados de forma a informar se tanto o caçador e o pato estão prontos a começar a partida.

A implementação deste módulo encontra-se em `serial_port.c`, `serial_port.h`, `uart_macros.h`.

4. Estrutura e Organização do Código

4.1. **bitmap.c**

Este ficheiro contém as funções que nos permitiram carregar, desenhar e eventualmente apagar bitmaps. O código não é da nossa autoria, sendo este proveniente do blog do ex-aluno Henrique Ferrolho:

<http://difusal.blogspot.pt/2014/09/minixtutorial-8-loading-bmp-images.html>

No entanto adicionamos 2 funções ao referido código.

Aluno encarregue: João Maduro

Peso: 3%

4.2. **duck.c**

Este ficheiro contém todas as funções necessárias para o movimento dos patos (tanto os gerados por computador, como o pato controlado pelo utilizador).

Aluno encarregue: Fernando Alves

Peso: 10%

4.3. **extras.c**

Este ficheiro contém algumas funções necessárias ao funcionamento dos vários menus, bem como funções que fazem subscribe e unsubscribe a todos os periféricos.

Aluno encarregue: João Maduro

Peso: 10%

4.4. **game.c**

Este ficheiro contém as funções relativas à lógica de jogo em ambos os modos.

Alunos encarregues: João Maduro (40%) e Fernando Alves (60%)

Peso: 26%

4.5. **keyboard.c**

Este ficheiro contém as funções que desenvolve-mos na aula laboratorial sobre o keyboard.

Aluno encarregue: João Maduro (50%) e Fernando Alves (50%)

Peso: 3%

4.6. **main.c**

Este ficheiro contém a função main, que inicializa o menu inicial.

Aluno encarregue: João Maduro

Peso: 1%

4.7. **menu.c**

Este ficheiro contém as várias funções "menu".

Aluno encarregue: João Maduro (50%) e Fernando Alves (50%)

Peso: 10%

4.8. **mouse.c**

Este ficheiro contém as funções que desenvolve-mos na aula laboratorial sobre o mouse.

Aluno encarregue: João Maduro (50%) e Fernando Alves (50%)

Peso: 3%

4.9. **points.c**

Este ficheiro contém as funções que desenvolve-mos na aula laboratorial sobre o keyboard.

Aluno encarregue: João Maduro (30%) e Fernando Alves (70%)

Peso: 10%

4.10. rtc.S

Este ficheiro contém as funções para obter os valores dos registos do RTC.

Aluno encarregue: Fernando Alves

Peso: 3%

4.11. serial_port.c

Este ficheiro contém as funções para comunicação entre computadores.

Aluno encarregue: João Maduro

Peso: 15%

4.12. timer.c

Este ficheiro contém as funções que desenvolvemos na aula laboratorial sobre o timer.

Aluno encarregue: João Maduro (50%) e Fernando Alves (50%)

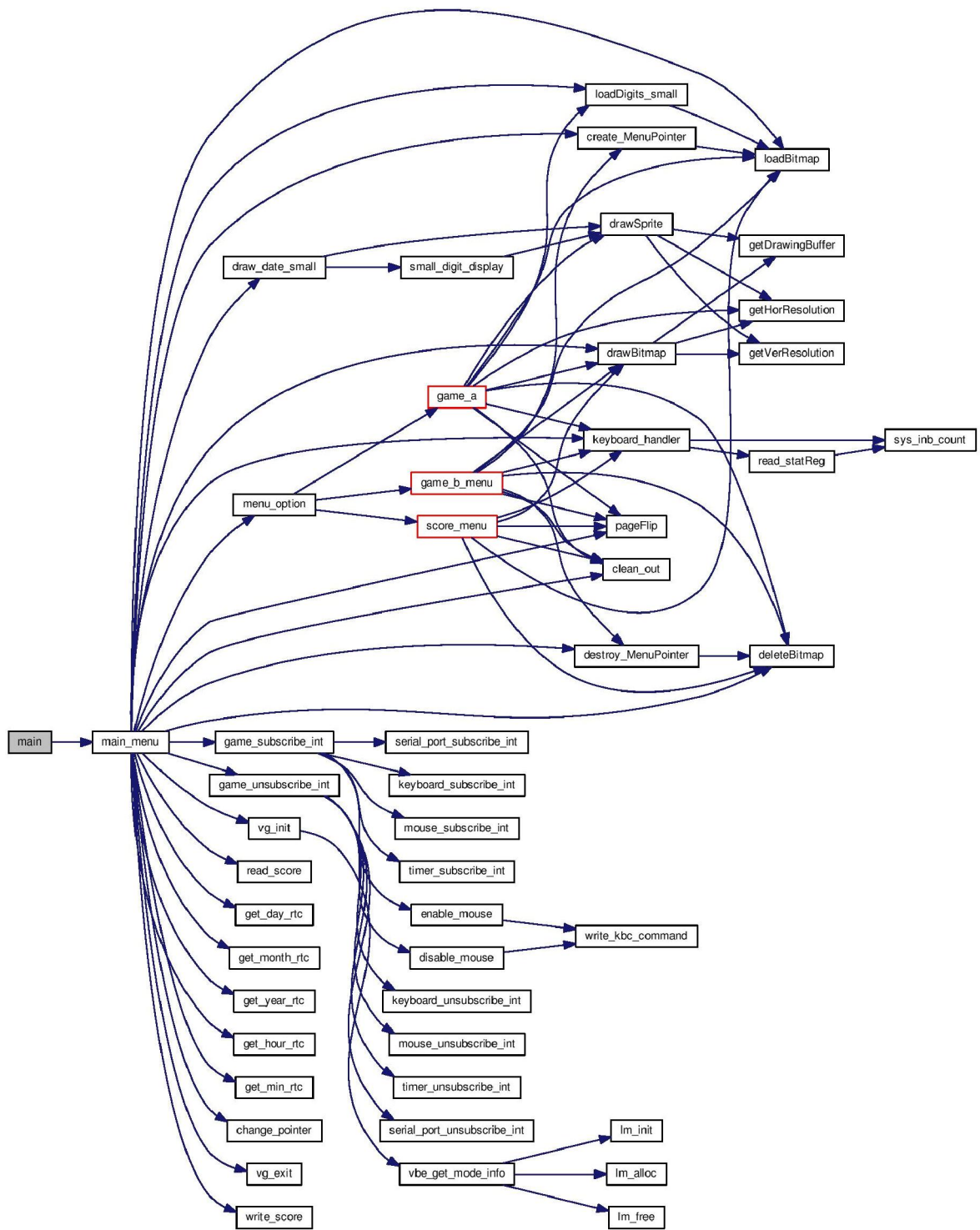
Peso: 3%

4.13. vbe.c e video_gr.c

Este ficheiro contém as funções que desenvolvemos na aula laboratorial sobre a graphics card.

Aluno encarregue: João Maduro (50%) e Fernando Alves (50%)

Peso: 3%



5. Detalhes de Implementação

5.1. Porta de Série

Durante o desenvolvimento do projeto deparamo-nos com alguns problemas durante a implementação da porta de série, nomeadamente em compreender como funciona o envio e receção de dados, uma vez que inicialmente as interrupções não eram acionadas quando recebia dados. Outro problema com que nos deparamos foi a ocasional não receção de dados muito cruciais para o funcionamento do jogo. Problemas de sincronização também foi um dos desafios que enfrentámos durante o desenvolvimento do projeto. Sendo este um módulo cujo lab não trabalhámos influenciou a dificuldade na sua aprendizagem. Resumidamente este periférico mostrou-se bastante complicado de perceber e implementar, mas ainda assim foi-nos possível arranjar uma implementação funcional para aquilo que pretendíamos.

6. Conclusão

Após a realização deste trabalho, achamos que agora entendemos muito melhor sobre os vários módulos lecionados ao longo do semestre e percebemos muito melhor a sua importância nos vários programas que utilizamos no dia-a-dia. Não só aprendemos a utilizá-los melhor, como a combiná-los e fazê-los trabalhar em conjunto para desempenhar diversas funções. Com este projeto também aprendemos a melhorar as nossas competências, enquanto programadores e adquirir novos conhecimentos nomeadamente, em linguagem C, código assembly, Orientação de Objetos e trabalhar com apontadores e referencias.

Nós achamos que esta cadeira é muito importante quer para aprender sobre os vários periféricos e a sua importância para a tecnologias modernas, quer para desenvolver a nossa capacidade de programar e abrir novos horizontes; mostrando-se, portanto, uma experiência muito positiva.

7. Apêndice

Para utilizar o projecto, primeiro extrai-se para a localização /home/lcom. Após a extração, é necessário mudar para modo superuser para termos permissões para os processos que o jogo precisa para funcionar. Para isso faz-se: “su”.

Para instalar o jogo é necessário correr o script de instalação com “sh install.sh”. Depois compila-se o código com “sh compile.sh”. Por fim, corremos o jogo com “sh run.sh”. Esperemos que se divirta