

CBALL

Relatório

FEUP

Projecto LCOM 2015-16

Turma 04

Grupo 03

Bruno Barros - up201405249

Fábio Caramelo - up201404783

Índice

Introdução	2
Instruções para o utilizador	3
Menu principal	3
Sobre o jogo	5
Estado do projecto	6
Periféricos utilizados	7
Descrição das funcionalidades de cada periférico	7
Timer	7
Teclado	7
Rato	8
RTC	8
Video	8
Organização de código	9
Sprite	9
Jogo	9
Timer	10
Keyboard	10
Mouse	10
Video	11
Menu	11
RTC	11
Diagrama de chamada de funções	13
Detalhes da implementação	14
Desenho do XPM	14
Double buffer	14
Máquina de estados	14
Constantes simbólicas	14
Sistema de colisões	14
RTC	15
Assembly	15
Conclusão	16
Observação	16
Contribuição	16

Introdução

O trabalho aqui apresentado foi realizado no âmbito da disciplina de LCOM, do 1º semestre do 2º ano do MIEIC da FEUP.

Realizamos este trabalho com o objetivo de o apresentar como projeto final da disciplina, então realizamos o trabalho destinado ao sistema operativo MINIX com base em C e tentamos usar o máximo de periféricos que conseguimos, nomeadamente o timer, o rato, o teclado, o video e o RTC.

O nosso jogo consiste numa representação do jogo Haxball mas numa versão de 1v1 com a possibilidade de controlar os dois jogadores com o teclado ou um dos jogadores com o teclado e o outro com rato, sendo este um jogo em multiplayer.

Instruções para o utilizador

Estando no directório que contém o código fonte do jogo, o utilizador deverá compilar o jogo utilizando o comando *make*. De seguida, devem ser concedidas permissões de root utilizando o comando *su*. Antes de executar o programa, deverá ser copiado o ficheiro *proj* contido na pasta *conf* para o directório */etc/system.conf.d*. Por fim, para executar o programa deverá ser executado o comando “*service run `pwd`/proj*”.

Menu principal

Seguindo as instruções previamente dadas, é apresentado ao utilizador o menu principal. O menu terá o seguinte aspecto:

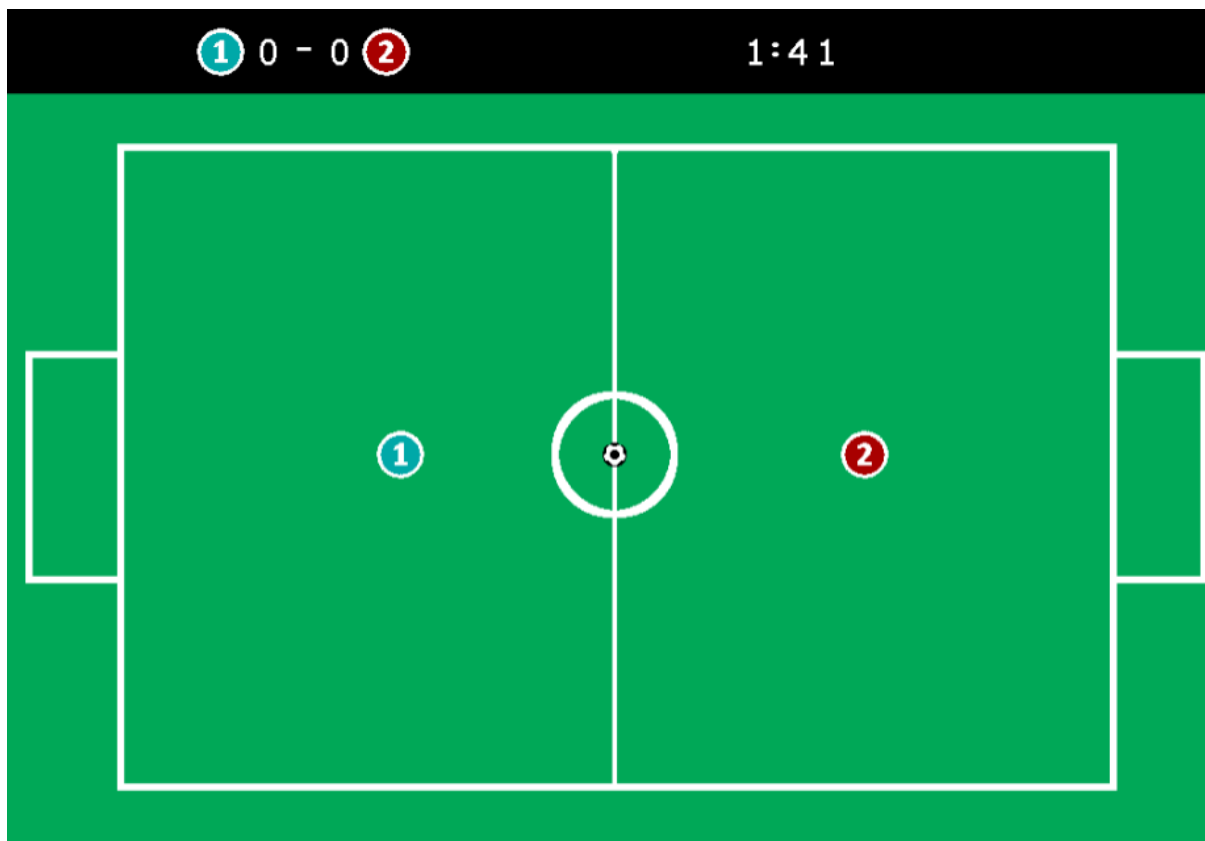


A navegação no menu pode ser feita através do rato ou das setas do teclado. Caso a opção seleccionada seja a opção do limite de golos, de tempo ou do modo de jogo, as opções podem ser alteradas premindo as setas para a esquerda e para a direita do teclado ou clicando com o rato nas setas visíveis no ecrã. Caso a opção seleccionada seja “Play” ou “Exit”, é possível seleccionar a opção clicando na tecla Enter ou clicando no botão esquerdo do rato.

- Opção **Play** - esta opção permite ao utilizador jogar. O jogo irá acabar quando o tempo tiver esgotado (tempo definido no menu) ou quando um dos jogadores tiver marcado o limite de golos (limite também definido no menu) e após o seu término, irá regressar ao menu.
- Opção **Time** - esta opção permite definir o tempo que irá limitar a duração do jogo. É possível escolher um limite de 2, 3, 4 ou 5 minutos de jogo clicando nas setas visíveis no ecrã ou premindo as setas esquerda ou direita.
- Opção **Goals** - esta opção permite definir o limite de golos no jogo. Quando um jogador atingir o limite de golos definido o jogo irá ser dado como terminado, regressando ao menu principal. É possível não existir um limite de golos, sendo o jogo dado como terminado apenas no final do tempo de jogo. É possível definir o limite de golos como 3, 4 ou 5.
- Opção **Mode** - esta opção permite definir o modo de input do jogo. Caso o modo de jogo seleccionado seja apenas de teclado um jogador pode mover-se com as teclas das setas e a tecla “.”(ponto) para rematar e o outro pode mover-se com as teclas “W A S D” e o seu remate é feito com a tecla de espaços. Caso o modo de jogo seleccionado seja teclado + rato, o jogador 1 será controlado com as teclas das setas e com a tecla “.” (ponto) para rematar e o jogador 2 será controlado com o movimento do rato e o seu remate será com o botão esquerdo do rato.
- Opção **Exit** - esta opção permite sair do jogo regressando ao sistema operativo.

Sobre o jogo

O cenário de jogo é bastante semelhante a um campo de futebol. O campo de jogo é constituído por duas balizas e tem forma rectangular. O objectivo do jogo é marcar o máximo de golos possível, inserindo a bola na baliza do adversário. Cada vez que a bola ultrapassa na totalidade a linha de golo da baliza, os jogadores a bola são colocados de novo das posições iniciais e é acrescentado um golo ao jogador que inseriu a bola na baliza adversária.



O campo de jogo é o representado na figura abaixo:

Estado do projecto

Todos os pontos acima referidos foram cumpridos. Contudo, sentimos que há aspectos do jogo que poderiam estar melhor, nomeadamente: O motor de jogo não está como desejaríamos pois por vezes existem colisões que não são tratadas da melhor maneira, isto é, por vezes os jogadores entram um dentro do outro ou dentro da bola, provocando comportamentos anormais nos movimentos. Não conseguimos melhorar este aspecto pois perdemos muito tempo até conseguirmos ter o jogo no estado que o temos uma vez que nunca tínhamos feito um projecto semelhante assim como foi o nosso primeiro desenvolvimento de um jogo e não tínhamos experiência com este tipo de problemas. Foi bastante desafiante construir o jogo.

Na especificação do projecto tínhamo-nos proposto a utilizar o RTC para guardar recordes ou fazer um registo de jogos, contudo limitamo-nos a utilizar o RTC para apresentar a data actual no ecrã inicial, isto é, no menu.

Gostaríamos também de ter utilizado outro modo de vídeo para incluirmos a utilização de bitmaps pois sentimos que algumas imagens não têm uma boa qualidade. A meio do desenvolvimento do projecto ainda implementámos outro modo de vídeo para utilizar bitmaps e conseguimos mostrar bitmaps no ecrã, contudo o movimento dos jogadores não funcionava da maneira pretendida. Como não conseguimos resolver o problema, reverteremos para o modo de vídeo original (0x105) e continuámos a utilizar XPM.

Periféricos utilizados:

Periférico	Utilização	Interupções
Timer	Controlo de frames por segundo, controlo de tempo de jogo e de inactividade do rato	Sim
Teclado	Navegação no menu, controlo de um ou dois jogadores no jogo	Sim
Rato	Navegação no menu, controlo de um jogador no	Sim

	modo de jogo “teclado + rato”	
RTC	Visualização da data no menu principal	Sim
Vídeo	Visualização do programa (menu e jogo)	Não

Descrição das funcionalidades de cada periférico:

- **Timer** - o timer permite controlar o número de frames por segundo do jogo, assim como controlar o limite de tempo do jogo e a animação da bola. As funções correspondentes ao timer encontram-se nos ficheiros timer.h e timer.c e são utilizadas no proj.c.
- **Teclado** - o teclado permite em primeira instância movimentar o utilizador no menu, podendo assim alterar as definições de jogo, como o tempo de jogo e o número limite de golos nesse jogo, permite também dar início ao jogo e também sair do ecrã de jogo. O teclado também permite controlar quer seja um, quer sejam dois jogadores, um com as setas, outro com as setas “W A S D”, o primeiro caso em que se joga com o teclado e o rato, o segundo quando se utiliza apenas o teclado. As funções correspondentes ao teclado estão presentes nos ficheiros keyboard.c e no keyboard.h e são executadas no proj.c.
- **Rato** - o rato permite alterar a opção seleccionada no menu, alterar os limites e o modo de jogo e entrar no jogo ou terminar o programa. Dentro do jogo, caso o modo de jogo seleccionado seja de teclado + rato, é possível mover o jogador correspondente através do movimento do rato e chutar a bola através do clique do botão esquerdo do rato. As funções correspondentes ao rato estão presentes nos ficheiros mouse.c e mouse.h sendo utilizadas no proj.c.
- **RTC** - o RTC permite apresentar no ecrã principal a data e hora atual, assim como o dia da semana. Utilizamos o RTC por interrupções a cada atualização do RTC. As funções dedicadas ao RTC estão definidas no RTC.c e RTC.h e são usadas no proj.c.

- **Video** - utilizámos o modo de vídeo para mostrar no ecrã todo o programa, isto é, o menu e o campo de jogo. O modo de vídeo utilizado foi o 0x105, podendo utilizar um total de 256 cores. De modo a garantir uma melhor fluidez no jogo utilizámos dois buffers, sendo um deles o buffer da memória gráfica. Também possuímos sprites animadas, neste caso a bola de jogo que quando se encontra em movimento, mudamos o seu XPM, fazendo uma rotação, dando a sensação de movimento mais realista. A utilização do double buffer foi implementada nas seguintes linhas do proj.c: 88-99, 139-147, 193-198, 253-260, 271, 339, 340 e 481-518. A cada interrupção do timer desenhávamos no buffer auxiliar e depois o buffer era copiado para a memória gráfica, sendo apresentado no ecrã. Utilizámos apenas XPM's. Utilizámos a animação da sprite da bola com o auxílio do timer e detecção de colisões entre a bola e os jogadores ou entre os jogadores. As funções correspondentes ao modo de vídeo estão presentes no video.c, video.h, timer.c e timer.h (para impressão do tempo de jogo no ecrã) e as colisões e a animação da sprite estão nos ficheiros sprite.c e sprite.h, sendo utilizadas no proj.c

Organização de código

Para este trabalho tentámos separar ao máximo os módulos existentes, de forma a ficar melhor organizado, criamos então um .h e um .c dedicado a cada módulo, por exemplo “sprite.h” e “sprite.c”, “mouse.h” e “mouse.c” à excepção do módulo de jogo e do menu que estão incluídos no ficheiro proj.c.

Módulo “Sprite”

Este módulo tem por base a criação e movimentação das sprites criadas, ou seja mediante o tempo e a movimentação introduzida pelo utilizador faz movimentar a sprite.

Neste modulo são redefinidas as velocidades de cada sprite, permitindo a cada sprite acelerar e travar. Também é neste módulo que estão implementadas as colisões assim como o seu tratamento.

Responsável: Fábio Caramelo

Contribuição:

- Bruno Barros: 50 %
- Fábio Caramelo: 50 %

Importância para o projecto: 25 %

Módulo “Jogo”

Este módulo tem por base a criação de uma situação de jogo, previamente definida no menu, com um certo tempo máximo, um número máximo de golos e o modo de jogo, se é apenas jogado com o teclado ou se um dos jogadores joga com o teclado e outro com o rato.

Este módulo utiliza a função driver_recieve() de maneira a controlar tanto as interrupções do timer, como as do teclado e do rato.

Este módulo não tem um .c nem um .h específico e é definido juntamente com o módulo do menu no ficheiro proj.c.

Responsável: Bruno Barros

Contribuição:

- Bruno Barros: 50%
- Fábio Caramelo: 50%

Importância para o projecto: 15%

Módulo “Timer”

Este módulo é responsável pelo controlo das atualizações da memória de vídeo e também pelo controlo do tempo de jogo, da animação da bola, da velocidade dos jogadores, que faz com que eles acelerem e travem. Este módulo também subscreve as interrupções do timer.

Responsável: Bruno Barros

Contribuição:

- Bruno Barros: 50%
- Fábio Caramelo: 50%

Importância para o projecto: 10%

Módulo “Keyboard”

Este módulo é responsável pela interação do utilizador com o programa utilizando o teclado. Neste módulo é feita a leitura dos scan-codes das teclas premidas pelo utilizador assim como a subscrição e cancelamento das interrupções do teclado. Neste módulo a partir do pressionar das teclas de direção ou de remate, alteramos os dados de uma struct que vai depois proporcionar o movimento e remate dos jogadores.

Responsável: Fábio Caramelo

Contribuição:

- Bruno Barros: 40%
- Fábio Caramelo: 60%

Importância para o projecto: 15%

Módulo “Mouse”

Este módulo é responsável pela interação do utilizador com o programa. Neste módulo realiza-se a leitura dos packets enviados pelo rato, assim com a subscrição e cancelamento das interrupções do rato. Neste módulo possuímos uma struct de valores de movimentação e remate que são alterados através do movimento do rato e do pressionar do botão esquerdo do mesmo. Esta struct vai depois proporcionar o movimento do jogador.

Responsável: Bruno Barros

Contribuição:

- Bruno Barros: 60%
- Fábio Caramelo: 40%

Importância para o projecto: 15%

Módulo “Vídeo”

Este módulo é responsável pela inicialização do modo de vídeo, pelo desenho do tempo de jogo, do resultado e da data obtida do RTC assim como do regresso ao modo de texto.

Responsável: Fábio Caramelo

Contribuição:

- Bruno Barros: 50%
- Fábio Caramelo: 50%

Importância para o projecto: 10%

Módulo “Menu”

Este módulo é responsável pelo inicio do jogo, termino do programa e alteração das definições de jogo.

Este módulo utiliza a função `driver_recieve()` de maneira a controlar tanto as interrupções do RTC como as do timer, as do teclado e do rato.

Responsável: Bruno Barros

Contribuição:

- Bruno Barros: 50%
- Fábio Caramelo: 50%

Importância para o projecto: 5 %

Modulo “RTC”

Este módulo subscreve e ativa, por atualização, as interrupções do RTC.

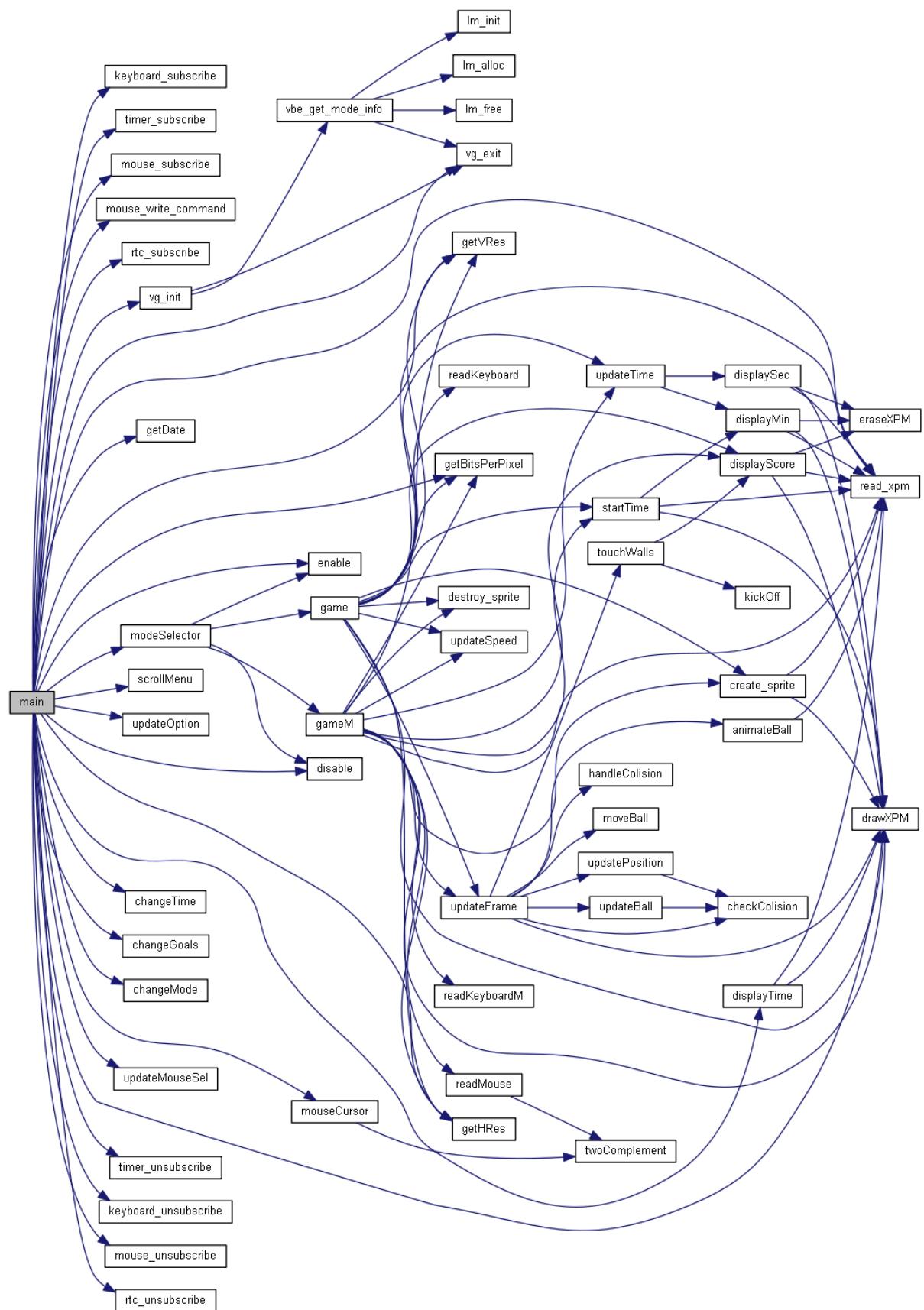
Responsável:

Contribuição:

- Bruno Barros: 50%
- Fábio Caramelo: 50%

Importância para o projecto: 5 %

Diagrama de chamada de funções



Detalhes da implementação

Neste projeto foi necessário implementar pequenos detalhes no código de maneira a que o programa pudesse ter uma apresentação e funcionamento melhor, mais adequado e também mais fluído.

Desenho dos XPM

De modo a permitir a existência de transparência, tivemos que modificar a função `draw_XPM` que tínhamos desenvolvido inicialmente nas aulas para ignorar uma cor (no nosso caso, cor 47 definida no ficheiro `const.h` como `IGCOL`).

Double buffer

Utilizamos este método de maneira a que fosse possível ter uma melhor imagem e uma melhor atualização da mesma, ficando o movimento de jogo mais fluído.

Máquina de estados

Utilizamos uma máquina de estados, no menu, de maneira a facilitar a navegação entre as opções disponíveis no menu. Foi criado um estado para cada uma das opções seleccionáveis.

Constantes Simbólicas

Utilizamos constantes simbólicas na maioria das situações de forma a facilitar que um mesmo valor seja aplicado em todas as aplicações.

Sistema de Colisões

O sistema de colisão foi definido para que fosse possível ao jogador ter contacto físico com outro jogador como também o drible do jogador com a bola, também só era possível o remate do jogador se a bola estivesse em contacto com o mesmo. Utilizámos o sistema de detecção de colisões entre dois objectos redondos leccionado na aula teórica do aluno Henrique Ferrolho.

RTC

Utilizámos o RTC para mostrar a data actual bem como o dia e a hora. Utilizámos as interrupções por actualização do RTC.

Assembly

Utilizámos assembly para converter os bytes que chegavam em complemento para 2 na leitura dos packet's do mouse. Este assunto foi abordado nas aulas, contudo, recorreremos a um [website](#) para complementar o conhecimento adquirido.

Conclusão

No final da unidade curricular concluímos que evoluímos muito pois tivemos que aprender e desenvolver o que nos era pedido num curto espaço de tempo. Ao longo do semestre fomos adquirindo conhecimento mais pormenorizado acerca dos periféricos abordados. Sentimos também que aprendemos a lidar melhor com matérias que nos eram desconhecidas num curto espaço de tempo e a lidar melhor com várias situações que tínhamos de resolver num curto espaço de tempo.

No entanto pensamos que existem pontos a melhorar, nomeadamente:

- por vezes a informação necessária à realização dos laboratórios encontra-se dispersa entre os slides das aulas teóricas e os guiões.
- o prazo de entrega dos laboratórios poderia ser fixo, isto é, independente dos horários das aulas laboratoriais.

Consideramos o acompanhamento do monitor nas aulas laboratoriais foi bastante útil e proveitoso, reduzindo bastante o tempo de espera para esclarecer dúvidas nas aulas.

Observação: na execução do programa nas nossas máquinas virtuais deparamo-nos com um problema na execução pela primeira vez sempre que se inicia a máquina virtual. Da primeira vez que executamos o programa o Minix não o consegue correr pois diz que não consegue alocar memória. Contudo, se comentarmos as linhas correspondente ao XPM do texto do menu (linhas 343, 482 e 527), compilarmos e correremos o programa ele executa. Deste modo, submetemos o código com as linhas referidas comentadas para que não haja problema na execução do programa. Após a primeira execução do programa as linhas podem ser descomentadas e o programa poderá ser compilado de novo para que seja executado de forma ideal.

Contribuição no relatório:

- Bruno Barros: 50%
- Fábio Caramelo: 50%