

Faculdade de Engenharia da Universidade do Porto



Relatório de Projeto - Pongix - T3G13

Ana Silva – up201505505 – up201505505@fe.up.pt
Danny Soares – up201505509 – up201505509@fe.up.pt

Relatório de Projeto realizado no âmbito da unidade curricular Laboratório de Computadores do
Mestrado Integrado em Engenharia Informática e de Computação

30/12/2016

Índice

1-Instruções de utilização	3
2-Estado do projeto	5
3-Organização e estrutura do código	10
Gráfico de chamada de funções	11
4-Detalhes de implementação	12
5-Avaliação da unidade curricular	13
6-Instruções de instalação	14

1. Instruções de utilização

O programa apresenta uma interface simples e apelativa, que permite aos utilizadores desvendar o que é necessário para nele se orientarem, abrindo com um menu inicial, no qual tem um cursor, que é movido usando o rato, para seleccionar uma de duas opções, jogar (“Play”) ou sair (“Exit”).



Fig. 1 - Ecrã de Menu Inicial

Na figura acima é possível ver a aparência do menu inicial, onde se vêem as opções e o cursor, sob a forma de um quadrado verde, de forma a distinguir-se do fundo preto e das figuras brancas.

Se o utilizador escolher a opção “Exit”, o programa encerra, voltando ao terminal do Minix.

Caso a opção escolhida seja “Play”, é apresentado o ecrã de jogo, no qual cada utilizador controla uma barra, utilizando as teclas “w” e “s” para fazer subir e descer a barra da esquerda, respetivamente, e as teclas “o” e “k” para fazer o mesmo com a barra da direita.



Fig. 2 - Ecrã de Jogo

Os utilizadores podem jogar durante 60 segundos, após os quais o ecrã de jogo é substituído pelo ecrã de final de jogo, onde aparece qual o utilizador que venceu ou se o jogo terminou empatado. O tempo de jogo aparece no topo do ecrã, centrado, iniciando em 60 e decrementando até 0. Ao lado esquerdo e ao lado direito do tempo surgem, além da data e da hora, os identificadores dos utilizadores das barras respectivas.

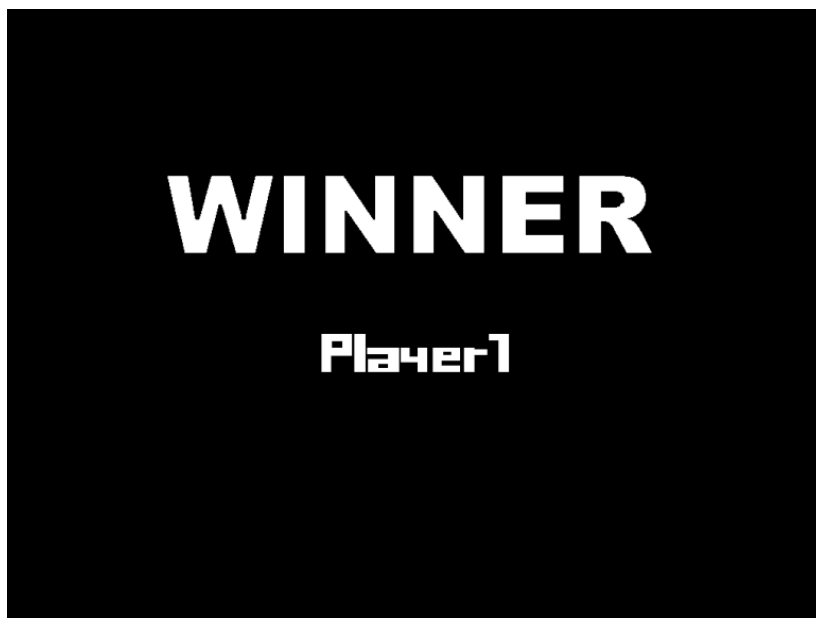


Fig. 3 - Ecrã de Final de Jogo

Caso não queiram completar os 60 segundos, ao pressionar a tecla “ESC”, os utilizadores avançam para o ecrã de final de jogo. Passados 2 segundos, o programa volta ao menu inicial.

2. Estado do Projeto

Chegando ao fim o tempo de desenvolvimento do projeto, foram cumpridos todos os objetivos para o programa, exceto implementar o módulo da porta série. Como tal, os periféricos abordados nas aulas laboratoriais, “timer”, teclado, rato e placa gráfica, foram utilizados no projeto, juntamente com o “rtc”, que não foi abordado nas aulas laboratoriais.

Periférico	Função	Interrupções?
Timer	Controlar a velocidade dos elementos móveis do jogo; Temporizador do jogo.	Sim
Teclado	Mover as barras dos jogadores;	Sim
Rato	Mover o cursor no menu inicial para selecionar opção;	Sim
Placa Gráfica	Mostrar o menu e os seus elementos; Mostrar o jogo e os seus elementos;	Não Não
RTC	Mostrar a data e a hora;	Não

Todas as opções possíveis de selecionar no jogo funcionam, isto é, o módulo que não foi implementado não tem qualquer presença no programa.

2.1. Timer

O “timer” tem um papel importante neste projeto, sendo que os movimentos da bola e das barras estão dependentes deste periférico, sendo estes movimentos a parte mais importante do jogo. Esta dependência é visível na função `int game()`, na qual são tratadas as interrupções do periférico durante o jogo, fazendo mover os objetos e fazendo decrementar o tempo do jogo. Os movimentos da bola são tratados na função `void ball_movement()`.



Fig. 4 - Utilização do "timer" no jogo

Na Fig. 4 é possível ver, assinalados a azul, os elementos móveis do jogo, cujos movimentos são controlados pelo timer, e, assinalado a vermelho, o tempo de jogo, que inicia em 60 e termina em 0, decrementando 1 a cada 60 interrupções do timer, o equivalente a 1 segundo.

O periférico considerado neste módulo é também importante no menu inicial, sendo que o movimento do rodapé presente neste menu é controlado por si, o que é possível ver na função `int menu()`, onde a cada interrupção do “timer”, é feito um ligeiro movimento do texto em rodapé.



Fig. 5 - Rodapé no menu inicial

2.2. Teclado

A utilização do teclado no “Pongix” é importante para jogar, sendo a sua função mover as barras de cada jogador, de forma a tentar vencer o adversário. Para tal, os utilizadores pressionam as teclas associadas a cada movimento e quando essas teclas são premidas é ativado o movimento das barras, até que as teclas sejam soltas. A verificação do estado das teclas é feita na função `int game()` e o movimento das barras é tratado na função `int movement()`.

```
int movement(Bar *b){
    if(b->press == 1){
        if (b->scancode == KEY_W || b->scancode == KEY_O){ //up movement
            if((b->ya - b->speed) <= 170){
                b->ya = 171;
            }
            else
                b->ya -= b->speed;
        }
        if (b->scancode == KEY_S || b->scancode == KEY_K){ //down movement
            if((b->ya + b->ht + b->speed) >= 767){
                b->ya = 767 - b->ht;
            }
            else
                b->ya = b->ya + b->speed;
        }
    }
}
```

Fig. 6 - Tratamento dos movimentos das barras

Nos movimentos das barras é feita a verificação da posição após o movimento, de forma a limitar os seus deslocamentos ao espaço designado para o jogo. O mesmo acontece com a bola, que se move só no espaço designado para o jogo. Para evitar que algum destes elementos saia do espaço de jogo ou que a bola passe por cima ou através de alguma das barras, são tratadas as colisões entre os objetos, de forma a que quando a bola atinge um objeto, é “impulsionada” pelo objeto de volta, alterando a sua direção e mantendo-se jogável.

2.3. Rato

O rato é usado no menu inicial do programa para mover o cursor e seleccionar uma das opções, para jogar ou sair do jogo. Os movimentos do rato são verificados na função `int menu()` e os movimentos do cursor são feitos na função `void updateCursor()`.



Fig. 7 - Cursor do menu inicial

Na Fig.7 é possível ver o cursor, um quadrado verde, assinalado com um círculo azul. A forma do cursor é devida ao estilo do jogo e para se assemelhar à bola, sendo que a sua cor é verde, por forma a distinguir-se das duas cores principais do programa, o preto e o branco.

Para verificar qual opção é escolhida, faz-se a verificação da posição do cursor a cada interrupção do rato, e quando este estiver dentro da zona da opção, delimitada por uma linha branca, se for pressionado o botão direito do rato, é selecionada a opção.

2.4. Placa Gráfica

A placa gráfica é o periférico com papel mais importante neste projeto, uma vez que sem as suas funcionalidades não seria possível realizá-lo. Este periférico é utilizado para exibir no ecrã, no modo gráfico 0x105 do Minix, com resolução 1024x768 e 64 cores, todos os elementos que aparecem no ecrã, quer no menu, quer no jogo, quer no ecrã de vencedor.



Fig. 8 - Menu inicial



Fig. 9 - Ecrã de jogo



Fig. 10 - Ecrã de vencedor

Nas 3 imagens acima é possível ver as 3 situações em que se tira proveito da placa gráfica. Os objetos mostrados pela placa gráfica estão todos guardados na forma de pixmaps no ficheiro "pixmap.h", tal como era feito nas aulas laboratoriais deste periférico. A exibição dos objetos é feita com recurso à função char *read_xpm() disponibilizada pelos professores, e à função void test_xpm_show() desenvolvida no trabalho laboratorial. Para utilizar a placa gráfica são utilizadas também as funções char *vg_init() e int vg_exit() do trabalho laboratorial. A placa gráfica é usada com double buffering, de forma a tornar mais eficiente a mudança de posição dos elementos no ecrã.

2.5. RTC

O "rtc" é utilizado neste projeto para mostrar a data (no formato dd/mm/aa) no menu inicial e para mostrar a data (no formato dd/mm/aa) e as horas (no formato hh:mm:ss) durante o jogo.



Fig. 11 - Exibição da data e da hora

O "rtc" é usado na função int menu_show(), para ser exibida a data no menu e é usado na função void ball_movement(), para exibir a data e a hora no jogo, por esta ser uma função que está constantemente a ser chamada durante o jogo, assim a hora está sempre a ser atualizada, de forma a mostrar os segundos corretos em cada momento.

2.6. Participação dos elementos do grupo

Módulo do "timer": a função que trata as interrupções do "timer" foi aproveitada do trabalho laboratorial. As situações em que as interrupções do "timer" são necessárias, para os movimentos ou decrementar o tempo de jogo, utilizam essa função e foram implementadas pelos dois elementos, durante as aulas laboratoriais de preparação do projeto, tendo sido feitas em conjunto.

Módulo do teclado: o tratamento das interrupções do teclado é feito com uma função aproveitada do trabalho laboratorial, sendo que foi necessário implementar uma função para reconhecer os "scancodes" das teclas utilizadas para jogar. Esta função foi implementada numa das aulas laboratoriais de preparação do projeto, pelos dois elementos.

Módulo do rato: o movimento do rato é tratado com a função desenvolvida no trabalho laboratorial, tendo sido aperfeiçoada pela Ana Silva, para corrigir um problema em movimentos com x ou y negativos. A verificação do estado do botão esquerdo do rato é feito com uma função desenvolvida no trabalho laboratorial.

Módulo da placa gráfica: as funções utilizadas para manipular a placa gráfica neste projeto foram aproveitadas do trabalho laboratorial, tendo sofrido ligeiras alterações, feitas por ambos os elementos do grupo, sendo portanto a participação dos elementos semelhante.

Módulo do “rtc”: a leitura da data e da hora é feita numa função desenvolvida pelo Danny Soares, enquanto que a exibição destas é feita em várias funções, desenvolvidas por ambos os elementos, de acordo com algumas necessidades.

Em geral, a participação dos elementos do grupo foi equilibrada, não havendo discrepâncias a apontar.

3. Organização e estrutura do código

3.1. Pongix (handler.h e handler.c) – 20%

Este é o principal módulo do projeto, no qual está a função “main” e é feita a chamada das funções principais para correr o programa. É neste módulo que estão as funções “menu” e “game”, as quais são o centro de processamento das ações dos utilizadores durante a execução do programa.

3.2. Placa Gráfica (graphics.h e graphics.c) – 20%

No módulo da placa gráfica, estão as funções responsáveis por mostrar todos os elementos que surgem no ecrã durante a execução do programa, recorrendo, em muitos dos casos, a pixmaps guardados no ficheiro pixmap.h. Este é um dos módulos mais importantes por ser o que permite que o programa seja exibido no ecrã.

3.3. Teclado (keyboard.h e keyboard.c) – 17%

O coração deste programa é o jogo, a dois jogadores, sendo importante, para tal, ter forma de mover as barras dos jogadores, o que é feito utilizando interrupções do teclado. Este módulo é também muito importante, por ser o que permite aos utilizadores jogar. Neste módulo estão implementadas as funções que tratam dos movimentos das barras dos jogadores, assim como o movimento da bola de jogo e o do rodapé que surge no menu inicial, tratando as colisões entre elementos da forma adequada.

3.4. Rato (mouse.h e mouse.c) 15%

O rato tem pouca atividade no programa, apesar de a sua atividade ser importante para jogar ou fechar o programa. No módulo do rato estão definidas as funções que permitem movimentar o cursor que aparece no menu inicial, usando os movimentos do rato do utilizador. Neste módulo também está definida a função que verifica em todas as posições do cursor, se está na área em que pode selecionar uma opção do menu.

3.5. Pixmaps (pixmap.h) – 10%

Os elementos que são mostrados no programa são maioritariamente pixmaps, por exemplo o título do jogo, os nomes dos jogadores, etc, sendo exibidos com as funções implementadas no módulo da placa gráfica.

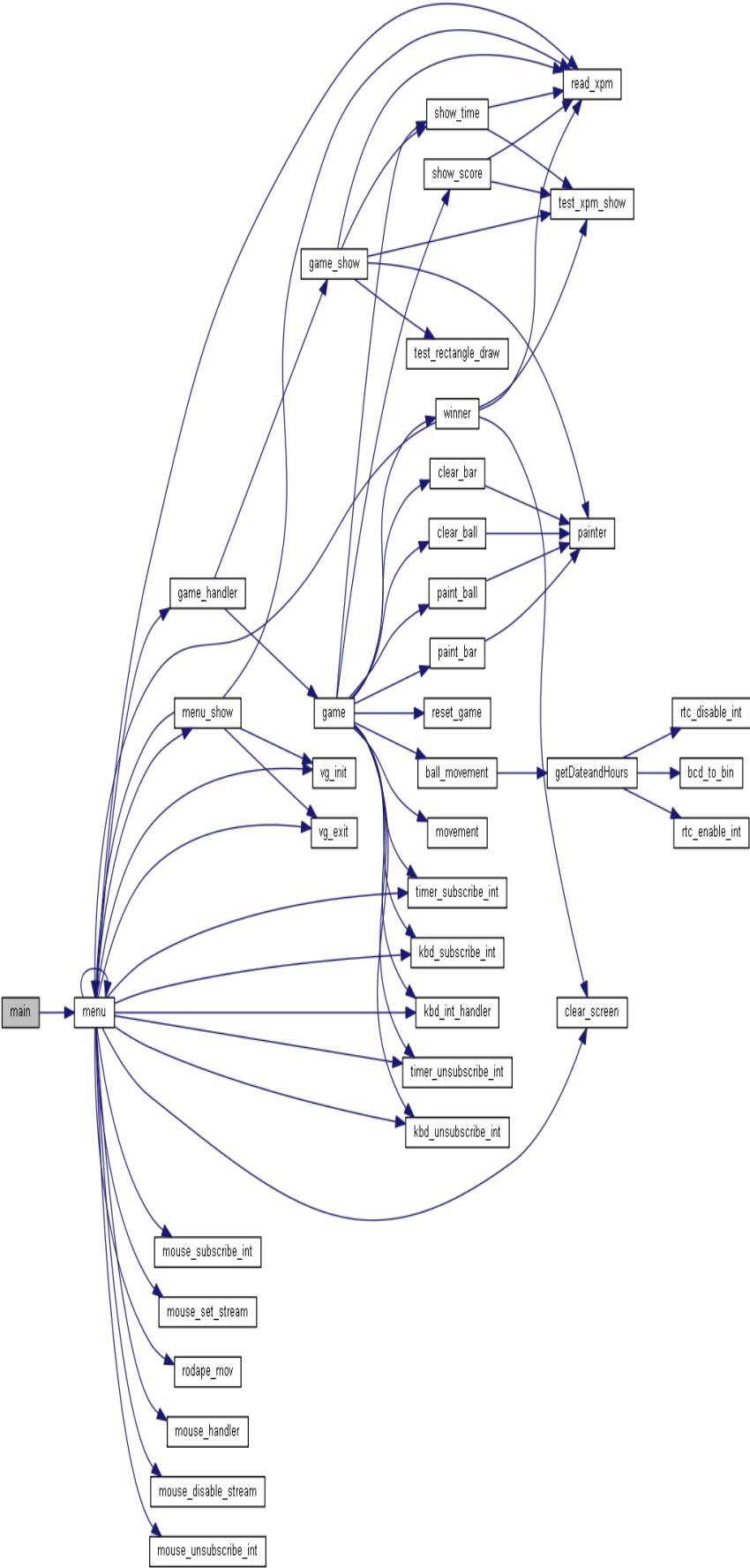
3.6. RTC (rtc.h e rtc.c) – 8%

O módulo do rtc é responsável por mostrar a data e a hora aos utilizadores.

3.7. Timer (timer.h e timer.c) – 10%

O timer é utilizado para controlar a velocidade dos movimentos dos objetos, através das suas interrupções, controlando a taxa de atualização do ecrã.

Gráfico de chamada de funções



4. Detalhes de implementação

Este projeto foi realizado com base nos trabalhos desenvolvidos nas aulas laboratoriais, tendo sido feitas adaptações sempre que necessário, de forma a tornar as funções mais eficientes.

Os módulos referidos na secção anterior foram desenvolvidos de acordo com os conhecimentos que nos foram passados nas aulas teóricas e nas aulas laboratoriais, utilizando os periféricos mais comuns dos computadores, através de funções desenvolvidas na linguagem de programação C e algumas em assembly, e recorrendo a conceitos como máquinas de estado ou orientação a objetos.

Tendo em conta o objetivo principal de um jogo de pong, foram implementadas as funções que permitem mover as barras dos jogadores utilizando as teclas designadas. Porém, verificar o estado de cada tecla em cada interrupção pareceu-nos mais eficiente implementar utilizando uma máquina de estados, guardando o estado da barra, isto é, se alguma tecla para a mover está premida, assim como o seu scancode, caso esteja premida, de forma a saber se o movimento deverá ser ascendente ou descendente. Desta forma é possível ter as duas barras a mover ao mesmo tempo caso as suas teclas respetivas estejam premidas, efeito que não era possível apenas com as interrupções do teclado. Igualmente, no menu foi usada uma máquina de estados para verificar se o cursor está dentro da área de uma opção, e, caso esteja, é alterada uma flag no cursor, mudando de estado, para o estado em que ao pressionar o botão esquerdo do rato é possível seleccionar a opção.

Os elementos mostrados durante a execução do programa foram definidos na forma de pixmaps, dada a sua simplicidade, não sendo necessário guardá-los como bitmaps para depois serem utilizados. Dessa forma, foi também possível utilizar as funções das aulas laboratoriais que imprimem pixmaps, fazendo ligeiras alterações.

No projeto são usadas duas funções em assembly, com a função de ativar e desativar as interrupções do rtc. Estas funções são muito simples, alterando apenas o estado da interrupt flag para o estado requerido.

Os movimentos e a taxa de atualização do ecrã são controlados pelas interrupções do timer, sendo que a cada interrupção do timer é feita uma atualização. Desta forma, o programa funciona a 60fps (frames per second), sendo esta taxa adequada, pois não requer alterações à frequência do timer e permite movimentos fluídos sem quebras nem arrastamentos.

5. Avaliação da unidade curricular

Após um semestre foi-nos possível concluir que a unidade curricular LCOM aumentou bastante os nossos conhecimentos em relação a periféricos e à sua manipulação, aprendemos a programar a um nível mais baixo do que estávamos habituados, utilizando C e por vezes assembly, o que também contribuiu para aumentar o conhecimento em programação mista destas duas linguagens. Além do ganho de conhecimento, esta unidade curricular tem um aspeto muito positivo que são os resultados finais, ou seja, os projetos, que dão muita motivação aos alunos e são projetos muito interessantes.

Por outro lado, há também alguns aspetos negativos a apontar, sendo o principal a dificuldade sentida para completar os trabalhos laboratoriais a tempo do final da aula, visto a nossa aula laboratorial ser à terça feira, havendo muita diferença de tempo para se preparar entre as turmas com aula no início da semana e as turmas com aula no final da semana. Em alguns trabalhos este problema foi combatido, tendo sido alargado o prazo de entrega dos trabalhos, o que permitiu que os alunos tivessem mais tempo para se organizarem e terem mais tempo para preparar os trabalhos de LCOM e estudar para as outras unidades curriculares.

Concluindo, a unidade curricular é importante para a formação dos estudantes, apesar de o grau de exigência ser elevado.

6. Instruções de instalação

Para o funcionamento do programa é necessário, depois de o ter no Minix, copiar o ficheiro de configuração do projeto para o diretório `/etc/system.conf.d`, utilizando o comando `"cp conf/proj /etc/system.conf.d/"` em root. Após ser executado este comando é necessário executar `"make"` e depois `"service run `pwd`/src/proj"`. Todos estes comandos devem ser executados a partir da pasta raiz do projeto (proj).