

Faculdade de Engenharia da Universidade do Porto



Jogo das Damas

Francisco Tuna Andrade - up201503481@fe.up.pt

José Manuel Faria Azevedo - up201506448@fe.up.pt

lcom1617-t2g06

Relatório de Projecto realizada(o) no âmbito da disciplina
Laboratório de Computadores
Mestrado Integrado em Engenharia Informática e Computação

Regente: Prof. Dr. Pedro Alexandre Guimarães Lobo Ferreira Souto

2016/01/02

Índice

Introdução.....	3
Instruções de utilização	4
Estado do Projeto.....	8
Organização de código.....	11
Função Call Graph	14
Detalhes da Implementação	15
Conclusão.....	17

Introdução

Neste relatório pretendemos apresentar o nosso projeto final. Pretendemos mostrar as instruções de utilização, explicar as funcionalidades dos periféricos que implementamos, expôr a forma como dividimos o projeto em vários módulos e explicar quais foram os nossos maiores desafios. Por fim, daremos também a nossa opinião acerca desta cadeira, mostrando os seus pontos positivos e negativos.

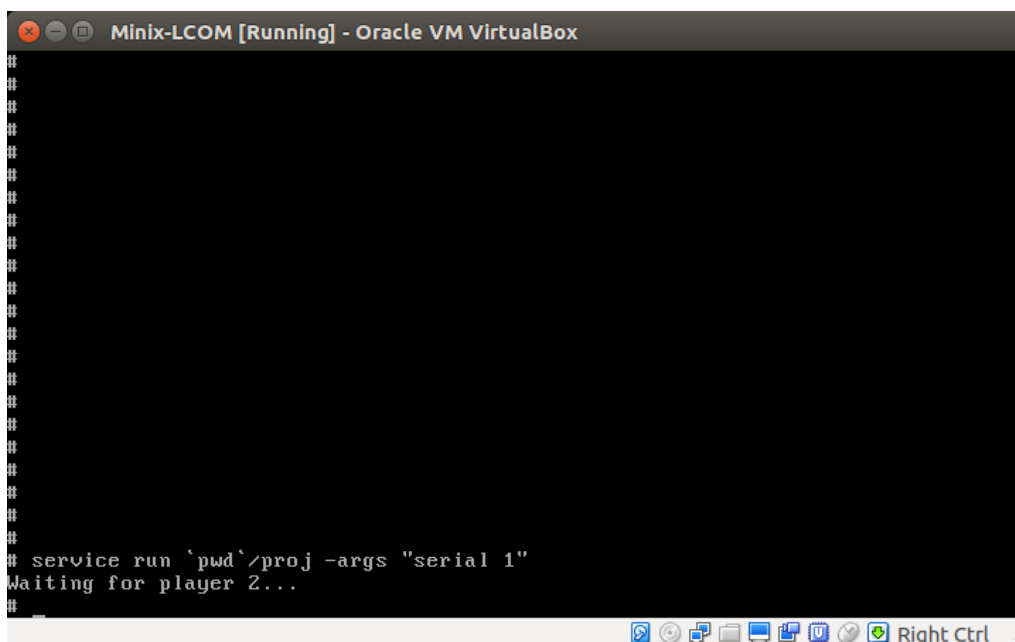
Instruções de utilização

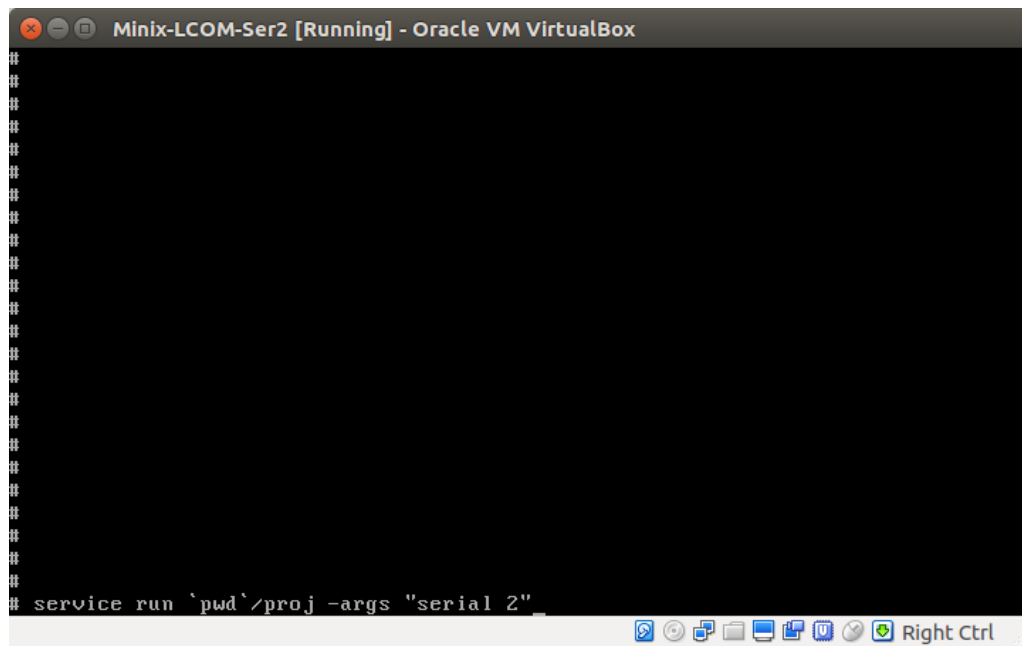
Iniciar o programa:

Este projeto tem dois modos, o modo *normal* onde é possível jogar um jogo de damas entre dois jogadores no mesmo computador e o modo *serial* onde os dois jogadores estão conectados através do serial port.

Para iniciar o programa no modo normal basta fazer service run dando como único argumento a palavra “normal”. Por exemplo um comando correto para executar o programa no modo normal será: `service run `pwd`/proj -args “normal”`

Para iniciar o programa no modo serial é necessário fazer service run dando dois argumentos: o primeiro é a palavra “serial” e o segundo é o número do jogador que pode ser “1”, se fôr o primeiro a jogar ou “2”, caso seja o segundo a jogar. É importante notar que o jogador 1 deve correr programa (com os argumentos “serial” e “1”) antes do jogador 2. Caso contrário, a conexão não será estabelecida. Na eventualidade de o utilizador enganar-se e correr o jogador 2 antes do jogador 1, ambas as máquinas virtuais onde os programas estão a ser corridos terão de ser reiniciadas ou a conexão não será estabelecida. Nas figuras abaixo pode-se observar o procedimento correto para iniciar o programa no modo serial:





Se correr o programa sem argumentos, é mostrado ao utilizador o procedimento correto para iniciar o programa

Publicidade:

Quando o utilizador inicia o programa no modo normal, ou seja, dois jogadores no mesmo computador, é-lhe mostrado um anúncio publicitário. O anúncio que é mostrado depende da data atual. Se essa data for um dia da semana (Segunda a Sexta), é mostrado um anúncio de uma cadeia de fast-food, enquanto que se for fim-de-semana é mostrado um anúncio de um filme cinema. Para fechar o anúncio e começar o jogo basta carregar na tecla ESC. Na figura abaixo pode ver-se um exemplo de um anúncio que é mostrado ao jogador num dia da semana:

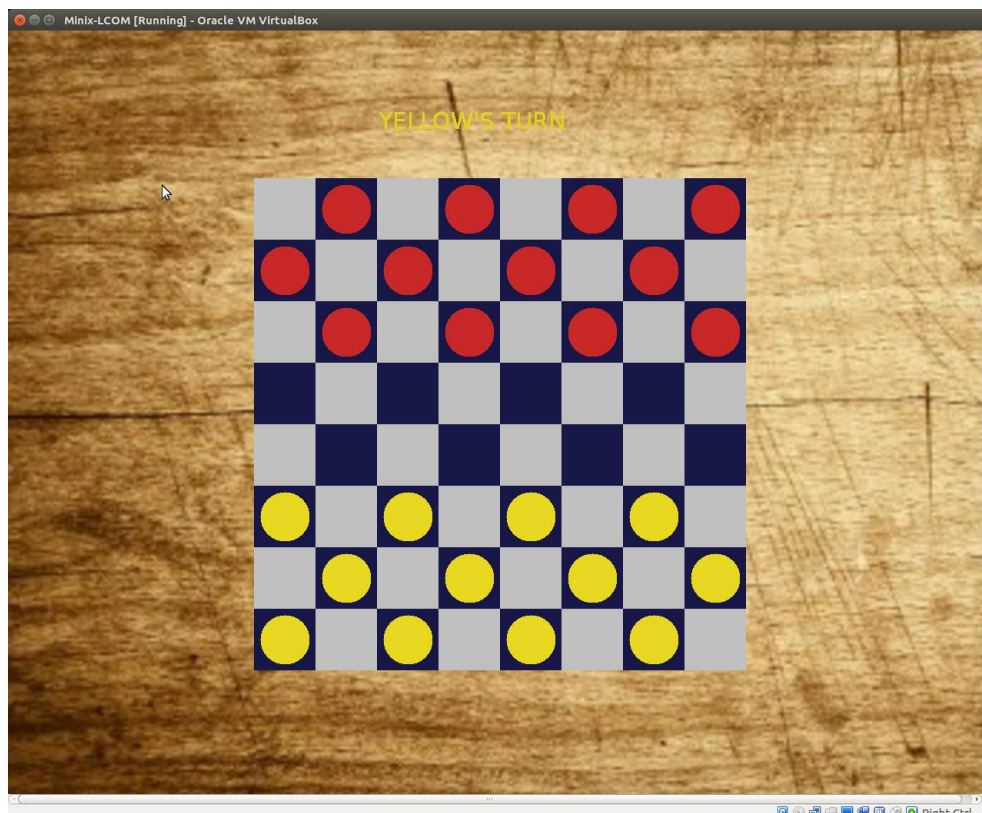


Jogo das Damas

Em relação ao jogo propriamente dito, as regras são as das Damas normais, ou seja, os jogadores são obrigados a comer uma peça adversária quando têm a oportunidade e apenas se podem mover para a frente uma casa para o lado esquerda ou uma casa para o lado direito. A única diferença em relação a um jogo normal é que o jogo acaba quando um jogador faz dupla dama, ou seja, quando chega ao fim do tabuleiro.

Para executar uma jogada basta selecionar a peça que se deseje com o rato e movê-la para uma posição válida. As peças movem-se com o rato quando o utilizador clica numa. No ecrã aparece uma mensagem que diz qual é o turno atual, se o das vermelhas, que se significa que deve ser o jogador com peças vermelhas a jogar, ou o das amarelas que significa que deve ser o jogador com as peças amarelas a jogar.

Cada jogador tem no máximo 30 segundos para cumprir uma jogada. Se ao fim de 30 segundos ainda não tiver efetuado nenhuma jogada, perde o jogo. Caso um jogador queira sair prematuramente basta selecionar a tecla ESC. Na figura abaixo pode-se observar o estado inicial do jogo:



Estado do Projeto

Funcionalidades dos periféricos:

Periférico	Funcionalidade	Interrupções / Polling
Timer	Contagem de tempo que cada jogador tem para executar uma jogada.	Interrupções
Keyboard	Saída do jogo	Interrupções
Mouse	Controlo das peças, execução de uma jogada	Interrupções
Graphics Card	Apresentação dos gráficos	Nada
RTC	Extração da data atual	Nada
Serial Port	Conectar dois jogadores no modo multiplayer	Polling

Descrição dos periféricos:

Timer

O timer é usado para cronometrar o tempo máximo que um jogador tem para executar uma jogada, sendo esta funcionalidade usada por interrupção. É também usada uma função `timer_set_square()` que no início do programa coloca a frequência do timer a 60. A implementação das funções do timer encontra-se no ficheiro `timer.c`, sendo elas chamadas a partir do ficheiro `checkers.c`.

Keyboard

O keyboard é usado para a saída prematura do jogo e para fechar os anúncios publicitários, funcionando por interrupções. É usada, nomeadamente, a função `kbc_cmd_read_ass`, escrita em assembly para a leitura de um scancode.

A implementação das funções relacionadas com o keyboard encontra-se no ficheiro `kbc.c` e no ficheiro `_kbc_ass.S`, sendo elas chamadas a partir dos ficheiros `checkers.c` para determinar quando é que um jogador quer sair prematuramente do jogo e a partir do ficheiro `publicity.c`, para determinar quando é que um jogador quer fechar um anúncio

Mouse

O mouse é utilizado para mover as peças de jogo, funcionando por interrupções. As funções mais importantes relacionadas com o mouse são o `mouse_convert_packet()` e o `mouse_get_packet()` que permitem ler e interpretar os packets enviados pelo mouse. Estas funções são usadas para obter a posição do mouse e para saber quando é que um jogador está a seleccionar uma peça.

A implementação das funções relacionadas com o mouse encontra-se no ficheiro `mouse.c` e elas são chamadas a partir do ficheiro `checkers.c`, para determinar as jogadas dos diferentes jogadores.

Graphics Card

O Graphics Card foi usado para mostrar os gráficos do projeto. Foram usados dois modos ambos RGB 565 o modo `0x117` com resolução `1024x768`, para mostrar os anúncios e o modo `0x11A`, com resolução `1280x1024`, para mostrar os gráficos relacionados com o jogo propriamente dito.

Foi usado triple buffering e também objetos em movimento, que eram o rato e as peças. Foi ainda usada a deteção de colisão entre o rato e os limites do ecrã, sendo que o programa não deixa o rato ultrapassar estes limites. Não foram usadas fonts mas foram usadas imagens com texto, que executam a mesma função.

As funções relacionadas com o Graphics Card encontram-se implementadas nos ficheiros vídeo_gr.c e no ficheiro vbe.c Bitmap.c, sendo chamadas através do ficheiro publicity.c para mostrar os gráficos relacionados com os anúncios e através do ficheiro checkers.c, para mostrar os gráficos relacionados com o jogo propriamente dito.

RTC

O RTC é usado para obter a data atual e assim mostrar um anúncio adequado ao jogador, que depende se fôr fim-de-semana ou não. É assim de chamar a atenção para a função rtc_read_date() que obtém a data atual do sistema.

As funções relacionadas com o RTC são implementadas no ficheiro rtc.c , sendo chamadas através do ficheiro publicity.c

Serial Port

O Serial Port foi usado para conectar dois jogadores no modo multiplayer, funcionando por polling. Quando dois jogadores se conectam no modo serial cada um deles envia e recebe uma match key. Caso a conexão tenha sido bem sucedida o jogo é iniciado. De seguida de cada vez que um dado jogador efetua uma jogada ele envia as coordenadas iniciais e finais da peça que moveu.

Os parâmetros de comunicação usados foram:

Bits per char: 8

Stop bits: 2

Parity: Par

Bits por segundo: 9600

É importante chamar atenção para as funções ser_send_char_poll() e ser_receive_char_poll(), que permitem enviar e receber um carater no modo polling, respetivamente. A implementação destas funções e de outras relacionadas com o serial Port encontra-se no ficheiro ser_port.c, sendo elas chamadas através do ficheiro proj.c para estabelecer a conexão entre dois jogadores e através do ficheiro checkers.c para enviar uma mensagem ao outro jogador de cada vez que uma jogada tenha sido concluída.

Organização de código

Módulos:

Proj (5%)

Este módulo contém o ficheiro main, sendo responsável por interpretar os argumentos dados pelo utilizador quando corre o programa e por estabelecer a conexão entre dois jogadores no modo serial

Checkers (25%)

É o módulo mais importante do projeto, responsável por lidar com todos os aspetos do jogo das damas propriamente dito, como por exemplo permitir que o jogador mexa o rato, permitir que o utilizador selecione uma peça e verificar se uma jogada efetuada pelo jogador é válida ou não.

ProgramLogic (20%)

Este módulo é utilizado para implementar as regras do jogo das damas. É importante notar que este módulo funciona a um nível mais abstrato do que o módulo anterior, isto é, ao contrário do anterior este módulo não lida com gráficos nem com outros periféricos, apenas implementa as funções que permitem por exemplo saber se uma jogada é válida ou não e que servem de suporte ao módulo anterior.

Publicity (4%)

Módulo utilizado para mostrar os diferentes anúncios publicitários, verificando qual o anúncio adequado dependendo da data atual.

Bitmap (5%)

Módulo responsável por lidar com os Bitmaps. Este código foi obtido na Internet no blog do antigo estudante da FEUP Henrique Ferrolho:

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

Foram feitas algumas modificações na função DrawBitmap() em relação à original, uma vez que a original desenhava um Bitmap no VideoMem enquanto que a nova função desenha um Bitmap no mouse buffer e no double buffer.

Utilities (2%)

Módulo com funções genéricas como por exemplo a conversão de um número em base 10 para base 16 ou a determinação do elemento máximo de um array.

Timer (3%)

Módulo responsável por implementar as funcionalidades do timer, que foram descritas na secção anterior.

KBC (6%)

Módulo responsável por implementar as funcionalidades do keyboard, que foram descritas na secção anterior.

_kbc_ass.S (2%)

Módulo em assembly responsável por implementar a função kbc_cmd_read_ass, que lê um scancode do keyboard.

Mouse (5 %)

Módulo responsável por implementar as funcionalidades do mouse, que foram descritas na secção anterior.

Video_gr / VBE (15%)

Módulo responsável por implementar as funcionalidades do Graphics Card, que foram descritas na secção anterior.

RTC (3%)

Módulo responsável por implementar as funcionalidades do RTC, que foram descritas na secção anterior.

Ser_port (5%)

Módulo responsável por implementar as funcionalidades do Serial Port, que foram descritas na secção anterior.

Distribuição dos módulos pelos elementos do grupo:

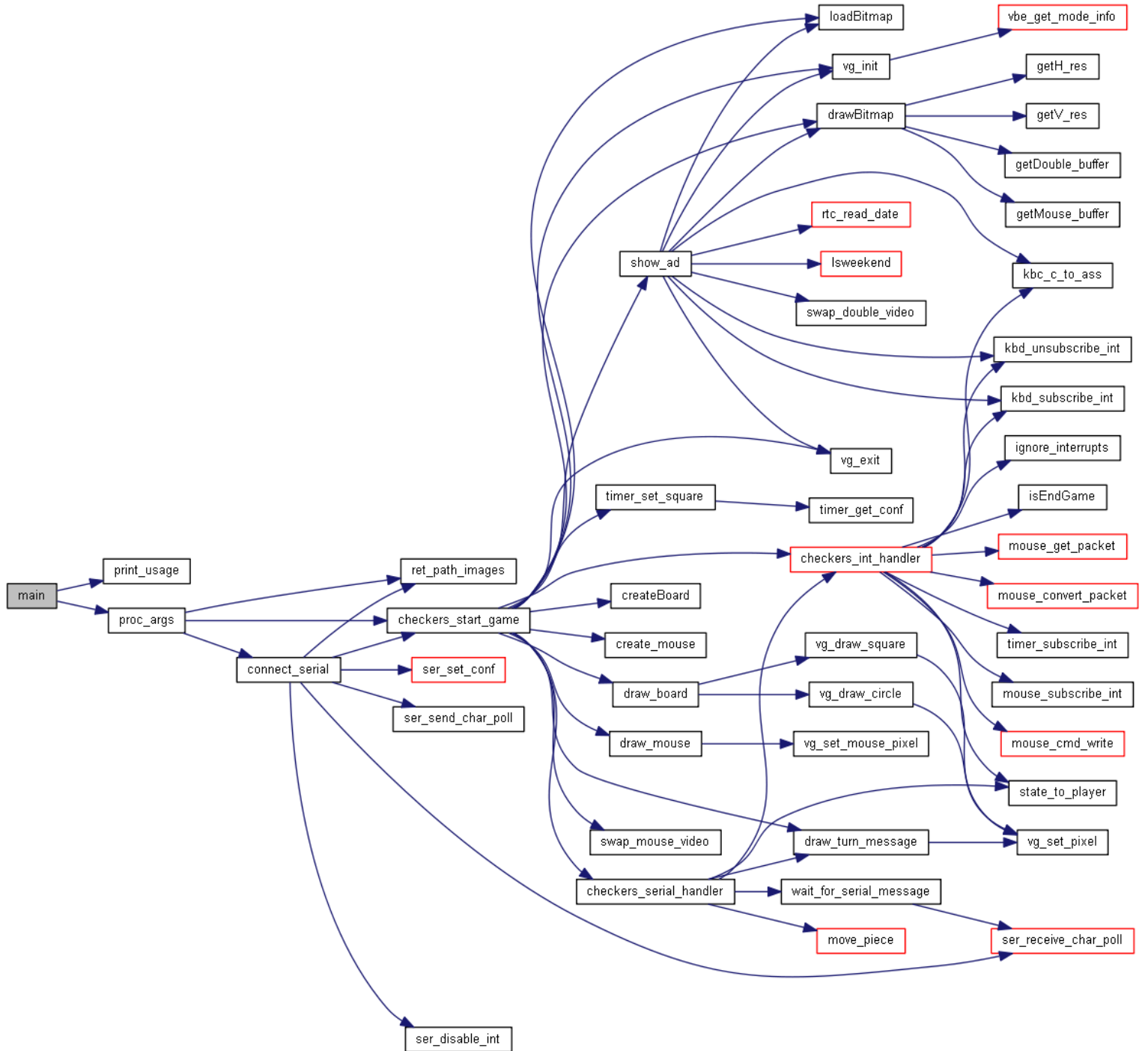
O estudante Francisco Andrade fez na sua totalidade os módulos: Proj, Checkers, Publicity _kbc_ass.S, Video_gr/VBE, RTC e Ser_Port.

O estudante José Azevedo fez na sua totalidade o módulo ProgramLogic.

Ambos os estudantes colaboraram e contribuíram nos seguintes módulos: Timer, KBC, Mouse e Utilities.

Conforme foi dito anteriormente, o módulo Bitmap foi, na sua maioria, obtido na Internet.

Função Call Graph



Detalhes da Implementação

A maioria das funcionalidades presentes neste projeto foram apresentadas nas aulas. Ainda assim, houve alguns detalhes que requereram um pouco mais de esforço e pesquisa para além do que foi dado nas aulas:

Assembly:

Foi especialmente cansativa a procura de uma solução para uma função em assembly que permitisse ler os scancodes emitidos pelo Keyboard. Isto porque a função que estudámos no lab3 usava uma variável global, algo que não era adequado neste projeto. Tivemos então de usar como parâmetro da função uma referência a uma variável scancode que registasse o scancode emitido pelo keyboard. Descobrimos então que não era possível converter uma referência em C para uma variável de memória em assembly, pelo que tivemos de usar uma função auxiliar `kbc_c_to_ass()` que convertesse uma referência em C para um pointer, que já pode ser devidamente interpretado em assembly. A obtenção de um scancode é portanto feita da seguinte forma:

```
unsigned long scancode;
...
kbc_c_to_ass(& scancode);
...
int kbc_c_to_ass(unsigned long *scancode) {
    return kbc_cmd_read_ass(scancode);
}
```

Mouse Buffering:

Exigiu também bastante esforço a implementação de uma solução que permitisse representar o rato do computador. A solução por nós encontrada envolveu o uso de triple buffering, sendo que o terceiro buffer, o mouse buffer era usado somente para os movimentos dos ratos. O nosso projeto teve ainda a dificuldade adicional de ter que mexer as peças em

conjunto com o rato e de determinar quando é que o jogador selecciona ou “desselecciona” uma peça.

Serial Port

Em relação ao Serial Port foi difícil confirmar que dois jogadores estão devidamente conectados e que um utilizador, por exemplo, não saiu do jogo e o outro jogador estava agora conectado a uma outra aplicação qualquer, ou que não estava conectado de todo.

Para isso de cada vez que um jogador envia mensagem pelo serial port, ele envia inicialmente um código, para que se possa ter a certeza que a o programa continua a correr conforme previsto.

Conclusão

Consideramos a disciplina de LCOM fundamental para o curso por nos fornecer uma introdução adequada aos periféricos do computador. Avaliamos o trabalho que desenvolvemos neste semestre de forma positiva e estamos satisfeitos com o resultado final, nomeadamente o projeto final.

Como sugestões deixamos o seguinte:

- Consideramos que todas as turmas deveriam ter o mesmo prazo de entrega para os labs, uma vez que como o ritmo das aulas é igual para todos, as turmas de sexta têm uma vantagem muito grande em relação às de segunda.

- O número de créditos dados por esta cadeira não corresponde ao trabalho que é necessário desenvolver. Esta cadeira é de longe a que dá mais trabalho do primeiro semestre, mas o número de créditos não indica isso.

- Deveria ter sido dada uma explicação mais elaborada acerca da linguagem assembly. Notamos que muitos estudantes tiveram dificuldade em executar as tarefas que dependiam de compreensão desta linguagem, fossem elas chamar o kernel através de assembly, chamar funções de assembly através de c...