

Faculdade de Engenharia da Universidade do Porto



ChessProject

Laboratório de Computadores



2016/2017 -- Mestrado Integrado em Engenharia Informática e Computação:

Turma 1, Grupo 1:

Afonso Bernardino da Silva Pinto
(up201503316@fe.up.pt)

Tomás Sousa Oliveira
(up201504746@fe.up.pt)

Docentes:

Pedro Souto pfs@fe.up.pt
Nuno Paulino nmcp@fe.up.pt
João Cardoso jmpc@fe.up.pt

Ricardo Nobre rjfn@fe.up.pt
Tiago Sousa tbs@fe.up.pt
José Pinto zepinto@fe.up.pt

Resumo

Criado no âmbito da disciplina Laboratório de Computadores leccionada no 2º ano do Mestrado Integrado em Engenharia Informática e Computação, ChessProject é o projeto final que combina todos os exercícios realizados durante o semestre.

Trata-se de um jogo de xadrez para dois utilizadores desenvolvido para o sistema operativo Minix 3.1.8.

Palavras-Chave

Xadrez; Minix 3.1.8; Laboratório de Computadores; FEUP.

Agradecimentos

Este projeto foi resultado de diversas contribuições e colaborações, dada de forma direta e indireta, mas todas elas essenciais à sua realização. Gostaríamos assim de expressar os nossos sinceros agradecimentos a todos os que tornaram possível este trabalho.

Ao professor Ricardo Nobre e ao monitor José Martins pela orientação dada e valioso acompanhamento constante durante o desenvolvimento do projeto.

Ao estudante Henrique Ferrolho pelo excelente tutorial que disponibilizou no seu blog com indicações para o desenvolvimento do projeto e pelos fragmentos de código que tornou acessível referentes à leitura de bitmaps.

Introdução

Este projeto foi realizado no âmbito da unidade curricular Laboratório de Computadores, do 2º ano do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

O objetivo deste projeto é implementar um jogo fundamentalmente em linguagem C, utilizando o maior número de periféricos possíveis dos desenvolvidos nas aulas da unidade curricular, nomeadamente, o Timer, Keyboard, Video Card e Mouse assim como periféricos que apenas foram abordados nas aulas teóricas como o RTC e a Serial Port.

Optamos por implementar um jogo de xadrez para dois utilizadores desenvolvido para o sistema operativo Minix 3.1.8. que não só recorresse à maioria desses periféricos como também reproduzisse fielmente as regras normais de um jogo de xadrez.

O presente relatório servirá para explorar as funcionalidades do jogo e fornecer detalhes sobre a sua implementação.

Índice

[Introdução](#)

[1. Instruções de Utilização](#)

[2. Estado do Projeto](#)

[2.1 Timer](#)

[2.2 Keyboard](#)

[2.3 Video Card](#)

[2.4 Mouse](#)

[2.5 RTC](#)

[3. Organização / Estrutura do Código](#)

[3.1 Bitmap 5%](#)

[3.2 Chess 20%](#)

[3.3 ChessProject 2%](#)

[3.4 Game 20%](#)

[3.5 Kbd 5%](#)

[3.6 Mouse 20%](#)

[3.7 RTC 10%](#)

[3.8 Timer 8%](#)

[3.9 Utilities 1%](#)

[3.10 Vbe 1%](#)

[4. Diagrama de Chamada de Funções](#)

[5. Detalhes de Implementação](#)

[5.1 Bitmaps](#)

[5.2 Implementação do RTC](#)

[5.3 Desafios nas Regras do Xadrez](#)

[5.4 Event Driven Code](#)

[5.5 State Machine](#)

[5.6 Object Orientation](#)

[5.7 Assembly](#)

[6. Avaliação da Unidade Curricular](#)

[7. Autoavaliação](#)

[8. Instruções de Instalação](#)

1. Instruções de Utilização

1.1 Menu Inicial

Ao correr o projeto é apresentado o seguinte menu:



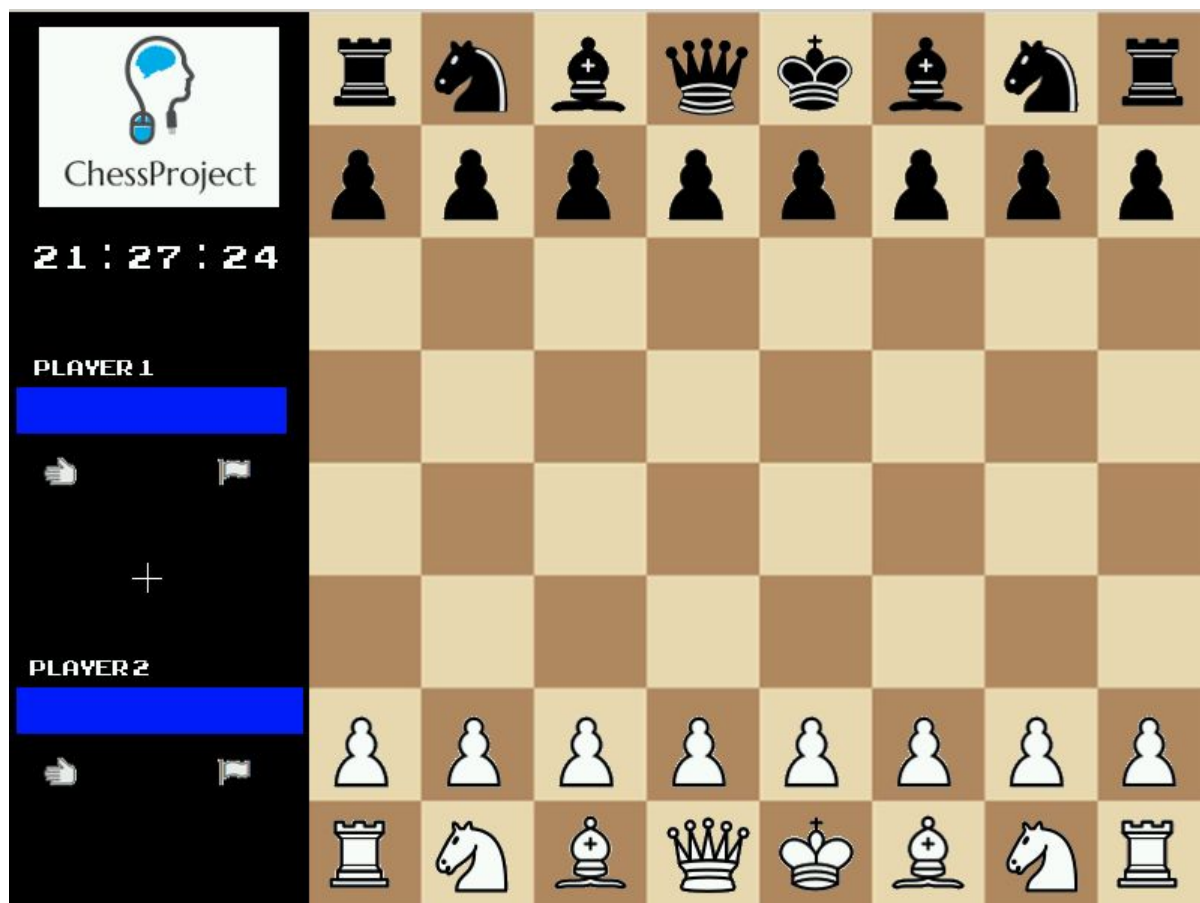
Além de mostrar ao utilizador a data e o dia da semana atual, este poderá escolher as opções “Multiplayer Local” (no qual o jogador poderá jogar contra um adversário no mesmo computador), “Multiplayer Serial” (no qual o jogador poderá jogar contra um adversário num computador diferente) ¹“Exit” (onde o utilizador poderá sair do jogo).

O controlo é feito através do rato.

¹ Lamentavelmente não foi implementado a tempo.

1.2 Multiplayer Local

Ao escolher o modo Multiplayer Local é apresentado o seguinte tabuleiro:



Aqui, além de ser apresentado o tabuleiro completo de xadrez, o utilizador terá ao seu dispor a hora atual, para ter uma noção do tempo enquanto joga. Através do rato, cada jogador poderá efetuar a sua jogada **clicando** na sua peça e de seguida no quadrado que o utilizador pretender, o movimento apenas surtirá efeito se estiver de acordo com as regras de jogo. O utilizador tem também a possibilidade de pausar o jogo, sair do jogo, pedir um empate ou desistir.

Como num jogo normal de xadrez as brancas é que começam, e logo depois as pretas, alternando cada rodada entre elas. É disponibilizado também o tempo (representado graficamente pela barra azul) que cada um tem para fazer a sua jogada, em que o “Player 1” corresponde às brancas e o “Player 2” às pretas.

Existem um total de 16 peças de cada cor, cada jogador tem ao seu dispor 6 peças diferentes (torre, bispo, cavalo, rainha, rei, peão). Quando uma peça “morre”, é retirada do jogo e este continua até um dos jogadores realizar o “Checkmate” ao outro, o jogo empatar através do “Stalemate”, o tempo de um dos jogadores acabar ou um dos elementos desistir.

Por fim, dependendo do resultado, irá aparecer no ecrã:



informando o vencedor.

O utilizador poderá abandonar esse ecrã premindo enter, regressando assim ao menu inicial onde poderá optar por realizar outro jogo ou sair.

2. Estado do Projeto

Todas as funcionalidades desenvolvidas ao longo do ano foram integradas no projeto e encontram-se a funcionar devidamente.

Para além dessas, também o RTC foi desenvolvido e utilizado.

Tínhamos como pretensão implementar ainda a Serial Port, tal como referido na proposta do projeto, contudo face a uma sobrecarga de projetos para a mesma data, não tivemos o tempo que seria necessário.

Periférico	Função	Interrupções?
Timer	Utilizado essencialmente para controlar o tempo que cada jogador possui para efetuar a jogada.	Sim
Keyboard	Utilizado tanto no menu como no jogo em si para pequenos atalhos.	Sim
Video Card	Utilizado para apresentar no ecrã os aspetos gráficos do jogo (menu, tabuleiro, imagens, texto...).	Não
Mouse	Utilizado para escolher as opções no menu e controlar as peças.	Sim
RTC	Utilizado para mostrar a data e a hora atual no ecrã.	Não

2.1 Timer

O timer é principalmente utilizado para gerir o tempo de jogo. Esta gestão é visível através de uma pequena barra que surge para cada um dos participantes, indicando graficamente (através da diminuição da barra), a passagem do tempo. As funções do timer estão implementadas nos ficheiros `timer.c` e `timer.h`, sendo usadas nas funções `game_management()` e `menu_management()`.

2.2 Keyboard

Apesar do Keyboard não ter sido muito usado, visto que o xadrez não necessita realmente da sua implementação para ser jogado, a sua presença é visível tanto nos menus como no jogo em si - para pausar (*Spacebar*), para sair (*Escape*) ou ainda andar uma jogada atrás (*Backspace*). As funções do keyboard estão implementadas nos ficheiros kbd.c e kbd.h e são usadas nas funções `waitForEnter()`, `mouse_write()`, `mouse_read()` e `mouse_int_handler()`.

2.3 Video Card

O Video Card foi usado para mostrar a parte gráfica do jogo, em modo gráfico. Utilizámos no modo 0x114, com resolução 800x600, com 64K números de cores e modo 5:6:5.

Foram usados 2 buffers para além da `video_mem` - buffer principal.

Todos os gráficos são desenhados no 2º buffer exceptuando o rato que é desenhado no terceiro buffer de forma a garantir uma melhor fluidez gráfica.

No desenvolvimento da interface gráfica do nosso jogo foram apenas utilizados bitmaps.

Não são utilizados sprites animadas, nem colisões.

As funções do Video Card estão implementadas nos ficheiros `vbe.c`, `vbe.h`, `video_gr.c`, `video_gr.h`, `bitmap.c` e `bitmap.h` e são usadas nas funções `chessproject_start()` e `chessproject_exit()`.

2.4 Mouse

O mouse é o principal dispositivo do nosso jogo, é usado tanto para escolher as opções do menu como também para controlar cada ação do jogo, especialmente designado para mover as peças. Para mover uma peça é necessário **clicar** na peça escolhida (com o botão do lado esquerdo do rato) e em seguida no quadrado para a mover.

As funções do mouse estão implementadas nos ficheiros `mouse.c` e `mouse.h` e são usadas nas funções `chessproject_start()`, `chessproject_exit()`, `game_management()`, `reset()` e `menu_management()`.

2.5 RTC

O RTC (Real Time Clock) é utilizado para mostrar a data e o dia da semana no menu inicial e a hora no modo Multiplayer. As funções do RTC estão implementadas nos ficheiros `rtc.c` e `rtc.h` e são utilizadas nas funções `game_management()` e `menu_management()`.

3. Organização / Estrutura do Código

3.1 Bitmap 5%

Este módulo possibilita o tratamento, a leitura, desenho e remoção de bitmaps, é essencial para a parte gráfica do nosso projeto. Cada imagem é do tipo “.bmp”.

Contribuição de cada elemento:

Afonso Pinto: 50% Tomás Oliveira: 50%

3.2 Chess 20%

Neste módulo está implementado as funções que permitem o funcionamento do jogo. Com este módulo os utilizadores podem jogar xadrez normalmente com as regras básicas e tradicionais do jogo.

Contribuição de cada elemento:

Afonso Pinto: 50% Tomás Oliveira: 50%

3.3 ChessProject 2%

Principal módulo do projeto com a função main, responsável por relacionar todos os outros módulos e organizar as operações de cada um.

Contribuição de cada elemento:

Afonso Pinto: 30% Tomás Oliveira: 70%

3.4 Game 20%

Módulo encarregue da manutenção do jogo e do menu. Este módulo é coração do projeto visto que é aqui onde estão localizadas as funções de gestão de jogo e menus - `game_management()` e `menu_management()`, que respetivamente, controlam o jogo e o menu, interligando outros módulos nelas.

Contribuição de cada elemento:

Afonso Pinto: 60%

Tomás Oliveira: 40%

3.5 Kbd 5%

Este módulo foi desenvolvido nas aulas laboratoriais e contém as funções necessárias para possibilitar a interação entre o teclado e o utilizador. O módulo é responsável por ler o código de algumas teclas, bem como a subscrição e cancelamento das interrupções no teclado.

Contribuição de cada elemento:

Afonso Pinto: 50%

Tomás Oliveira: 50%

3.6 Mouse 20%

Este módulo foi parcialmente desenvolvido nas aulas laboratoriais e contém as funções necessárias para possibilitar a interação entre o rato e o utilizador. É responsável pela leitura de informação e atualização do rato.

Contribuição de cada elemento:

Afonso Pinto: 60%

Tomás Oliveira: 40%

3.7 RTC 10%

Este módulo permite mostrar no ecrã a data, dia da semana e hora atual.

Contribuição de cada elemento:

Afonso Pinto: 55%

Tomás Oliveira: 45%

3.8 Timer 8%

Este módulo foi parcialmente desenvolvido nas aulas laboratoriais tem uma grande contribuição no jogo permitindo mostrar a cada jogador o tempo que tem para efetuar uma jogada e ditando o compasso para as atualizações do grafismo.

As funções de subscrição e cancelamento de interrupções foram necessárias para permitir mostrar o tempo.

Contribuição de cada elemento:

Afonso Pinto: 50%

Tomás Oliveira: 50%

3.9 Utilities 1%

Módulo encarregue de definir algumas estruturas importantes que se repetiam noutros módulos e assim evitar repetir código.

Contribuição de cada elemento:

Afonso Pinto: 30% Tomás Oliveira: 70%

3.10 Vbe 1%

Este módulo foi desenvolvido nas aulas laboratoriais, contendo as funções mais significativas para iniciar a placa gráfica, em modo gráfico.

Contribuição de cada elemento:

Afonso Pinto: 50% Tomás Oliveira: 50%

3.11 Video_gr 8%

Este módulo foi desenvolvido nas aulas laboratoriais, contendo funções que permitem desenhar retângulos e linhas, bem como inicializar a placa gráfica, em modo gráfico ou voltar ao modo texto.

Contribuição de cada elemento:

Afonso Pinto: 55% Tomás Oliveira: 45%

4. Diagrama de Chamada de Funções

5.1 Bitmaps

Criámos uma biblioteca de inúmeros bitmaps, do tipo “.bpm” usados para desenhar cada imagem do nosso projeto, incluindo as horas, a data, as peças, o tabuleiro, (entre outros..).

Devemos citar que o código que criámos foi feito com ajuda do website do estudante Henrique Ferrolho que devido aos tutoriais no seu blog, conseguimos perceber e implementar e modificar as funções do bitmap no projeto.

5.2 Implementação do RTC

Uma vez que não tivemos nenhuma aula laboratorial acerca deste tópico, foi necessário construir do 0 o módulo responsável por este dispositivo.

5.3 Desafios nas Regras do Xadrez

Utilizámos o ficheiro chess.c e chess.h para colocar as funções acerca das regras básicas e necessárias para um jogo normal do xadrez. No entanto, várias regras demonstraram-se difíceis de implementar, nomeadamente:

- O Bispo só pode andar diagonalmente;
- Criar o Roque (Castling), uma jogada especial do Xadrez, com o objetivo de proteger o Rei;
- Criar as Promoções do Peão quando ele chega à linha inicial do adversário, em que o peão poderá ser substituído por outra peça do tabuleiro;
- Criar o En Passant, um movimento de captura do peão;
- Criar o Check, CheckMate e StaleMate.

No entanto, conseguimos implementar todas, sem erros o que provou ser um desafio trabalhoso mas bem-sucedido.

5.4 Event Driven Code

Facilmente interligado com o tópico seguinte relacionado este paradigma de programação acompanha grande parte do nosso código para determinar as ações do utilizador, tais como clicks no mouse e no teclado.

5.5 State Machine

Foram implementadas várias máquinas de estado no projeto, especificamente:

- O estado do Menu, que mostra em qual dos menus o utilizador está a usar;


```
typedef enum {
    MENU,
    MULTIPLAYER_LOCAL,
    MULTIPLAYER_SERIAL,
    END
} MENU_STATE;
```

- O estado do Jogo, onde podemos saber qual dos jogadores é o próximo a jogar, se está no modo pausa e o resultado do jogo.

```
typedef enum {

    WHITE2PLAY,
    BLACK2PLAY,
    PAUSED,
    WHITEWINS,
    BLACKWINS,
    DRAW

} GAME_STATE;
```

- O estado do Movimento, onde podemos saber se o movimento é En Passant, Castling, um movimento normal ou nenhum movimento.

```
typedef enum {
    NOMOVE,
    NORMALMOVE,
    ENPASSANT,
    CASTLING
} MOVE_STATE;
```

- O estado dos Empates, onde podemos ver se ninguém pediu empate, ou se um dos jogadores pediu empate.

```
typedef enum {
    NONEDRAW,
    WHITEDRAW,
    BLACKDRAW
}DRAW_STATE;
```

5.6 Object Orientation

Não sendo claramente o forte da linguagem C, tentamos aproximar-nos ao máximo deste estilo de programação com o uso de Structs em diversos caso de particular realce no

módulo do mouse.

5.7 Assembly

Por fim usámos código assembly, mais concretamente, extended assembly, em várias funções no ficheiro game.c como por exemplo:

```
void winnerState(){
    if(game_state == BLACK2PLAY)
        asm ("movl %1, %%eax; movl %%eax,%0;"
            : "=r"(game_state)
            : "r"(BLACKWINS)
            : "%eax"
        );
    else if(game_state == WHITE2PLAY)
        asm ("movl %1, %%eax; movl %%eax,%0;"
            : "=r"(game_state)
            : "r"(WHITEWINS)
            : "%eax"
        );
}
```

Recorremos ao website GCC, the GNU Compiler Collection, para desenvolver e aprender mais sobre extended asm.

6. Avaliação da Unidade Curricular

Através desta unidade curricular, conseguimos familiarizar-mos-nos com a linguagem C e adquirir/cimentar conceitos sobre a interação com dispositivos. O carácter essencialmente

prático da disciplina é louvável e uma das características que mais prezamos. Contudo vários são os aspectos que poderiam ainda ser melhorados:

- Uma data fixa para a entrega dos trabalhos em detrimento do modelo atual;
- Alteração dos labs de ano para ano;
- Melhor organização dos guiões e slides;
- Correção dos labs não tão demorada;
- Fornecimento de Tutoriais para criar um ficheiros Doxyfile com diagramas;

7. Autoavaliação

Afonso Pinto

Contribuição: 50%

Participação: 50%

Tomás Oliveira

Contribuição: 50%

Participação: 50%

8. Instruções de Instalação

Para a execução do jogo foram criados shell scripts que seguem na pasta proj do repositório, eis como usá-los:

- su
- cd /home/lcom/proj
- sh install.sh
- sh compile.sh
- sh run.sh

Caso prefira pode colocar manualmente o ficheiro contido na pasta conf do diretório proj na pasta etc/system.conf.d (necessário permissões root), fazer make e service run `pwd`/ChessProject dentro da pasta proj.