

Faculdade de Engenharia da Universidade do Porto



KNIGHTS OF LCOM

**Projeto Final
Laboratório de Computadores
Turma 4 - Grupo 1**

Autores:

**Ângelo Rafael Araújo Oliveira (up202207798@up.pt)
José Pedro Pereira da Costa(up202207871@up.pt)
Bruno Miguel Ataíde Fortes(up202209730@up.pt)**

Índice

Introdução.....	3
1.Instruções de utilização do programa.....	4
1.1 Ecrã Inicial.....	4
1.2 Ecrã de escolha do modo de jogo.....	5
1.3 Carregamento de jogo previamente fechado.....	6
1.4 Instruções.....	8
1.5 Dentro do jogo.....	9
1.6 Menu de Pausa.....	11
1.7 Telas de vitória.....	12
2. Estado do Projeto.....	14
2.1 Tabela de funcionalidades:.....	14
2.2 Tabela de dispositivos:.....	14
2.3 Dispositivos utilizados.....	15
2.3.1 Timer:.....	15
2.3.2 Teclado:.....	15
2.3.3 Rato:.....	16
2.3.4 Gráfica:.....	17
2.3.5 RTC (Real Time Clock):.....	18
3. Estrutura e organização de código.....	19
Timer - 5%:.....	19
Teclado - 10%:.....	19
Rato - 10%:.....	19
Gráfica - 15%:.....	20
RTC - 5%:.....	20
Main - 10%:.....	20
Game - 20%:.....	20
Controller - 17%:.....	20
View - 5%:.....	21
Utils - 1%:.....	21
KBC - 2%:.....	21
3. Gráfico de chamada de funções.....	22
Conclusão.....	23

Introdução

O nosso projeto “Knights of LCOM” é uma adaptação do tradicional jogo de xadrez, desenhado para dois jogadores.

O jogo tem 3 opções, criar um jogo novo, carregar um jogo ou ler as instruções.

Na primeira opção , o usuário pode escolher entre vários modos de jogo (1min,5min,10min,30min,1hr), após escolher uma das opções o usuário é levado para dentro do jogo onde começa a jogar.

Na segunda, o programa vai buscar um jogo anteriormente guardado pelo usuário e vai retomá-lo.

Na terceira, o usuário pode entender o objetivo básico do xadrez e como deve proceder para deslocar as peças.

O objetivo do jogo consiste em fazer “checkmate” no rei inimigo, porém este pode acabar quando o tempo do contador de um dos jogadores acaba.

1.Instruções de utilização do programa

1.1 Ecrã Inicial



Fig 1. Ecrã Inicial

Ao iniciar o jogo, é exibido o ecrã inicial, com o nome do programa e três opções distintas, que podem ser seleccionadas com os botões do teclado que se encontram antes de cada instrução, essas opções são:

NEW GAME : Leva para a tela de escolha de modo de jogo.

LOAD GAME : Carrega um jogo anteriormente fechado, desde que o tempo do contador daquele jogador ainda não tenha acabado.

INSTRUCTIONS : Mostra as instruções de como deve ser jogado o jogo, além de uma pequena explicação sobre o objetivo do mesmo.

1.2 Ecrã de escolha do modo de jogo

Ao seleccionar a primeira opção no ecrã inicial, o usuário é direccionado para uma tela onde pode escolher entre diferentes modos de jogo: 1 minuto, 5 minutos, 10 minutos, 30 minutos ou 1 hora, seleccionáveis através do teclado. Para regressar ao menu inicial o usuário deve pressionar a tecla 1 do teclado.

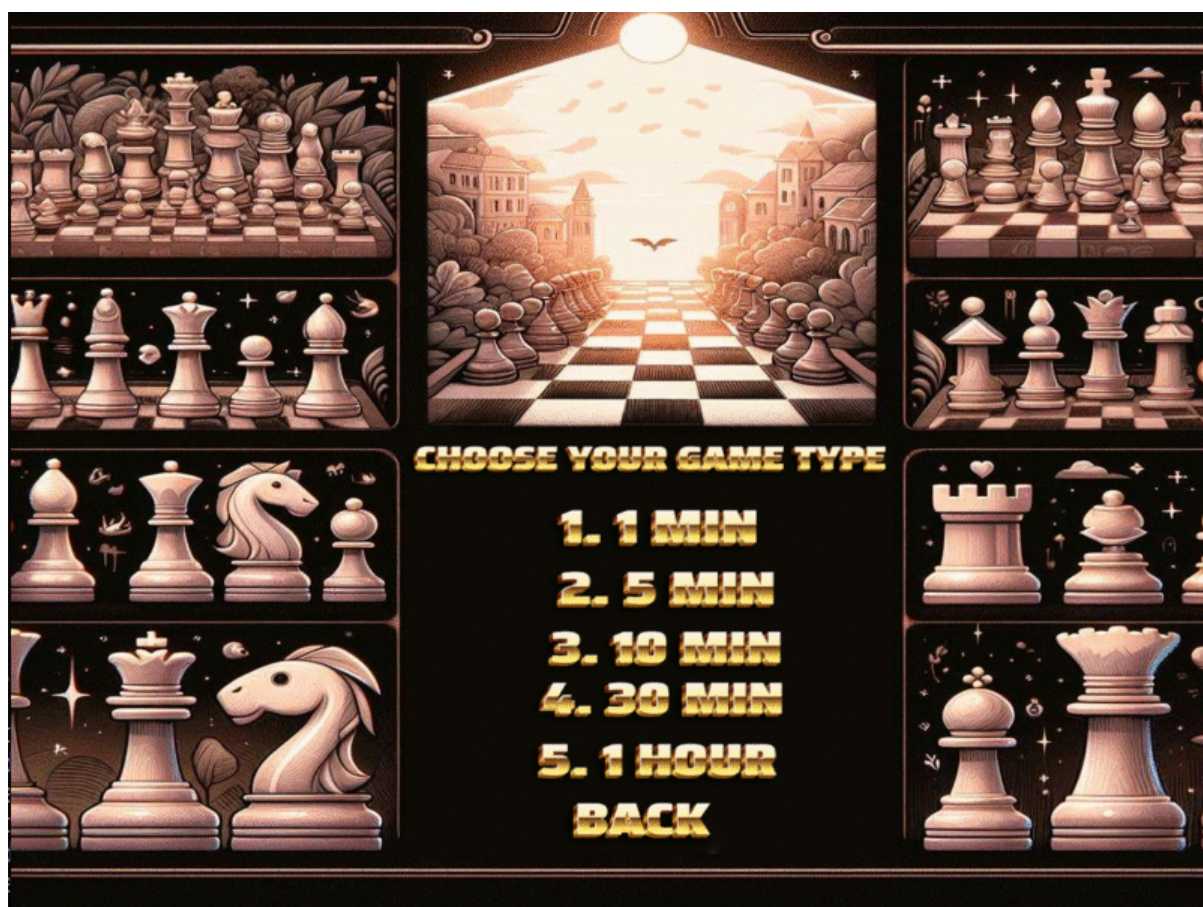


Fig 2. Ecrã de escolha do modo de jogo

1.3 Carregamento de jogo previamente fechado

Ao seleccionar a segunda opção no ecrã inicial, o usuário é direccionado para dentro de um jogo anteriormente aberto.

Este jogo pode ser acessado enquanto o programa se encontra aberto.

Caso nenhum jogo tenha sido iniciado anteriormente, o programa redirecciona para a tela mostrada anteriormente no ponto 1.2.

1.4 Instruções

Ao prosseguir na terceira opção do ecrã inicial, o usuário é levado para esta tela, onde está especificado as regras e objetivos do jogo, bem como deve proceder para mover as peças, para regressar ao menu inicial deve pressionar a tecla 1 do teclado.

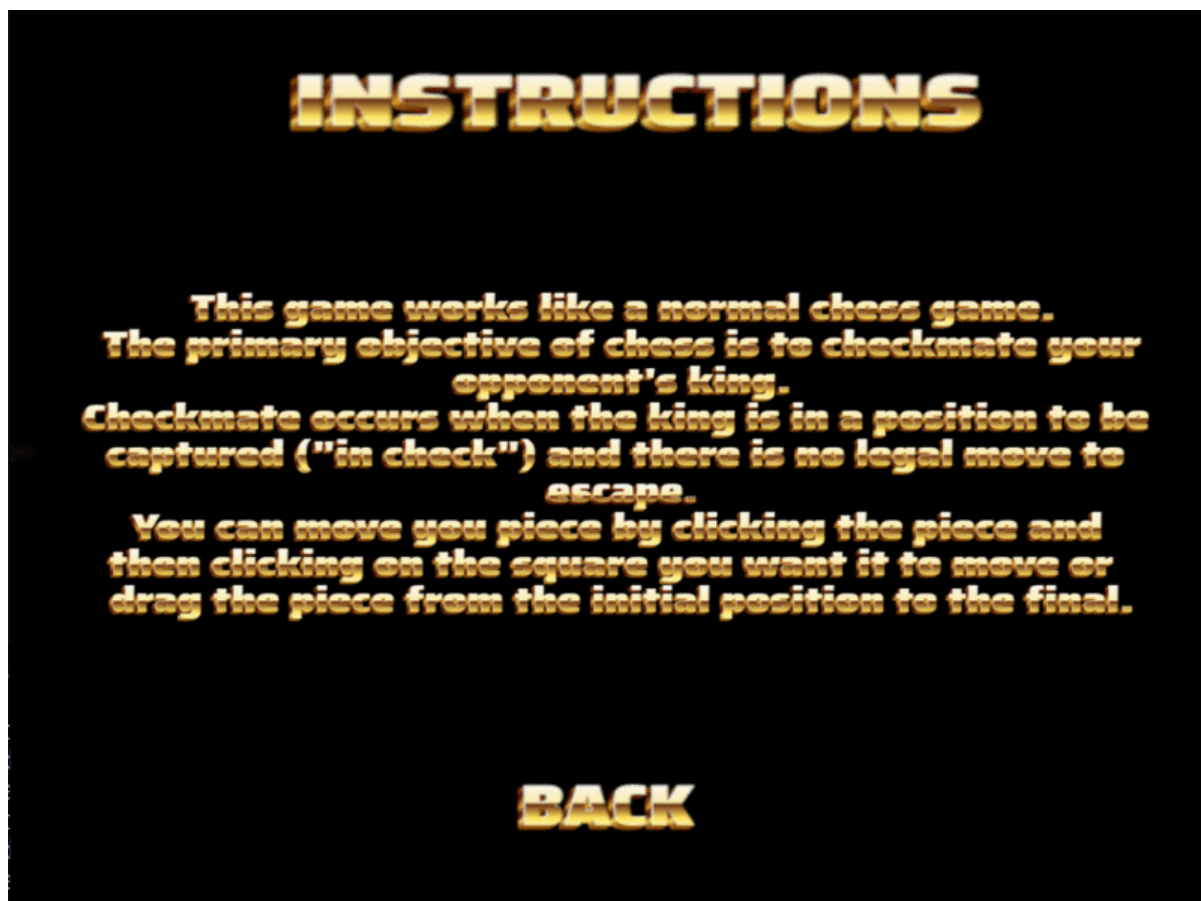


Fig 4. Instruções

1.5 Dentro do jogo

Após selecionar o modo de jogo, o usuário é levado para dentro do jogo, após isso o jogador que vai usar as peças brancas deve começar e fazer o seu primeiro movimento, através de um dos dois métodos:

- Clicar na peça com o botão esquerdo do rato que deseja mover e clicar no quadrado que quer que a peça se mova para.
- Clicar na peça com o botão esquerdo do rato e segurar, de seguida, deve arrastá-la para a casa que quer que a peça vá, e largar o botão esquerdo.

De seguida, o jogo prossegue de forma normal, com o jogador branco e o preto a alternarem os turnos, até que um deles perca o jogo.

Nesta tela, também é possível observar o tempo restante de cada jogador através de um contador, sendo o superior o dos pretos e o inferior o dos brancos, bem como um menu que pode ser acessado ao clicar em cima do texto “MENU”.



Fig 5. Dentro do jogo (Primeiro turno)



Fig 6. Dentro do jogo (Após alguns turnos)

1.6 Menu de Pausa

Ao clicar no botão de menu na tela de jogo, o usuário é redirecionado para uma tela de pausa, onde pode escolher voltar ao jogo (opção 1) ou voltar ao ecrã inicial (opção 2).

Qualquer um dos jogadores pode entrar no menu e o tempo do jogo que jogaria neste turno para de ser contado.



Fig 7. Menu de Pausa

1.7 Telas de vitória

Quando um dos jogadores ganha , seja por o tempo do adversário ter esgotado, ou por ter tomado o rei inimigo, o jogo termina e é levado para a tela de vitória, consoante a cor das peças do jogador que ganhou.

Nesta tela, é dada a opção de o usuário voltar ao menu inicial, pressionando a tecla 1.



Fig 8. Tela de vitória das peças brancas



Fig 9. Tela de vitória das peças pretas

2. Estado do Projeto

2.1 Tabela de funcionalidades:

Funcionalidades	Dispositivos Usados	Estado de Implementação
Navegar nos menus	Teclado, rato e gráfica	Completo
Mover as peças	rato e gráfica	Completo
Carregar jogo	RTC	Completo
Mostrar o tempo de cada jogador	Timer e gráfica	Completo
Desenhar peças e tabuleiro	gráfica	Completo
Visualizar o movimento anterior	Teclado e gráfica	Completo

2.2 Tabela de dispositivos:

Dispositivo	Usos	Interrupções
Timer	Mostrar o tempo restante de cada jogador dentro do jogo.	Sim
Teclado	Mover-se dentro dos menus, retroceder jogadas.	Sim
Rato	Mover uma peça, seleccionar o menu no jogo.	Sim
Gráfica	Trata de toda a interface do jogador, desde desenhar peças, desenhar menu, desenhar relógios	Não
RTC	Verificar se o tempo restante do jogador quando saiu da partida, desde que voltou a entrar já passou.	Sim

2.3 Dispositivos utilizados

2.3.1 Timer:

O timer é utilizado para contar o tempo restante de cada jogador quando dentro do jogo.

No nosso caso optamos por utilizar o timer 0, com uma frequência de 60 Hertz, garantindo assim que haja 60 interrupts por segundo.

A cada 6 interrupts é chamada a função `decrease_player_timer()`, que vai diminuir o contador de cada player em 1 décima de segundo, garantindo assim que após 60 interrupts, tenha sido decrementado 1 segundo.

A implementação do timer encontra-se no ficheiro `timer.c`, com os macros utilizados no ficheiro `i8254.h`.

2.3.2 Teclado:

O teclado é usado para navegar pelos menus e para visualizar um movimento anteriormente feito dentro do jogo.

O ciclo de vida de input, começa quando é detectado o primeiro interrupt do teclado e caso este não possua no primeiro bit 0xE0, então será o único para aquela tecla, caso possua, haverá dois interrupts.

O código útil que retiramos do teclado será apenas o make code (informação enviada quando o usuário pressiona uma tecla).

Em todos os menus, o usuário pode usar o teclado para seleccionar qual a opção que quer prosseguir, averiguando qual o número que aparece antes da opção. Ex: "1. New Game ",etc..

Além disso nas opções que consistem em "Back" ou "Exit", cujo número não aparece na tela, deverá ser pressionado sempre o número seguinte da última opção, cujo valor aparece na tela. Ex: "Exit" do menu inicial, é ativado quando pressionar a tecla 4.

Já dentro do jogo, ao pressionar a seta para a esquerda, o jogador voltará ao movimento anterior, ao clicar na seta para a direita, o jogador irá para o movimento seguinte (caso anteriormente tenha clicado para voltar para trás), na seta para baixo fará um "quick restart" voltando ao início do jogo e na seta para cima voltará à jogada atual.

A implementação do teclado encontra-se no ficheiro `keyboard.c`, com os macros utilizados no ficheiro `i8042.h`, partilha também do ficheiro `kbc.c` com o rato.

2.3.3 Rato:

O rato é usado para mover as peças dentro do jogo de xadrez ou para seleccionar o menu dentro do jogo.

Para conseguir usar o rato , o usuário deve ativar a detecção do rato na máquina virtual.

A posição do rato começa inicialmente no ponto (0,0) da tela e enquanto o usuário mexe o rato, o deslocamento deste origina a próxima posição.

O rato dentro do jogo tem uma lógica baseada numa máquina de estados, realizada através da função `in_game_mouse_movement()` com cinco estados:

- **INITIAL:** Em que o rato está à espera de um clique com o botão esquerdo numa peça válida ou no botão de menu, caso este seja feito numa peça passa para o próximo estado `PIECE_SELECTED`.
- **PIECE_SELECTED:** Após a peça ter sido clicada, neste estado ele vai verificar uma de duas hipóteses, ou o usuário permaneceu com o botão esquerdo clicado e avança para o estado `PIECE_DRAGGED`, o que significa que irá arrastar a peça ou o usuário largou o botão e significa que vai clicar na posição final.
- **PIECE_DRAGGED:** Espera até que o botão esquerdo seja largado para ver até onde a peça foi arrastada e avança para o estado `COMPLETE`.
- **PIECE_CLICKED:** Espera até que o botão esquerdo volte a ser pressionado, para saber onde terminar o movimento da peça ou espera que o usuário cancele o movimento apertando o botão direito do rato, caso a primeira opção avança para o estado `COMPLETE`, senão volta ao estado `INITIAL`.
- **COMPLETE:** Neste estado, o movimento foi concluído com sucesso e assim será processado para verificar a sua validade ou não, retornando ao estado `INITIAL`.

A implementação do teclado encontra-se no ficheiro `mouse.c`, com os macros utilizados no ficheiro `i8042.h`, partilha também do ficheiro `kbc.c` com o teclado.

2.3.4 Gráfica:

A gráfica é utilizada para desenhar os vários menus, as peças, a tábua, os contadores de tempo de cada jogador, etc...

Tudo que é desenhado pela gráfica é feito através de XPMs, originários de imagens .png, que foram cortadas, redimensionadas e alteradas através de softwares externos como o GIMP, ImageMagick e Mogrify.

A placa gráfica é configurado no modo VBE 0x115, com uma resolução de 800x600 pixels com cores RGB cada uma com 8 bits (1 byte), tendo assim em cada cor 3 bytes em modo direto, o que permite o programa ter uma gama de cores ampla, com cerca de 2^{24} cores diferentes que poderíamos escolher.

Para evitar problemas de fluidez no movimento das peças , na troca entre menus ou na mudança de tempo do contador dos jogadores, implementamos uma técnica de triple buffer por cópia, estando assim as imagens divididas por cada um dos seguintes buffers:

- **frontBuffer:** Este buffer é o que está a ser exibido atualmente na tela do jogador, é atualizado a partir do backBuffer após cada renderização, evitando falhas nas imagens.
- **backBuffer:** O back buffer é onde todas as operações de desenho ocorrem inicialmente. Durante cada ciclo de renderização, desenhamos todos os elementos gráficos no back buffer, incluindo peças, tabuleiro, menus e contadores de tempo, após este ciclo é copiado para o frontBuffer para ser apresentado, garantindo que o usuário nunca veja uma imagem incompleta ou em processo de atualização.
- **backgroundBuffer:** O background buffer contém o fundo estático o que permite uma atualização rápida do backBuffer, fazendo uma cópia do backgroundBuffer para o backBuffer antes de desenhar todos os elementos que podem conter movimentos.

Esta escolha foi feita, com o objetivo de garantir um jogo fluido visualmente e evitar problemas adjacentes como flickering (imagem na tela mudar durante o processo de atualização do monitor) e tearing (buffer ser desenhado antes de ser atualizado).

Além disso, para garantir que não seria preciso carregar o xpm de cada elemento em run-time, foi usado uma função xpm_load() que guarda em variáveis xpm_image_t , todos os XPMs necessário para o funcionamento do programa.

2.3.5 RTC (Real Time Clock):

O RTC é utilizado para contabilizar o tempo que cada jogador passa fora do tabuleiro de modo a combater qualquer tentativa de batota. Cada vez que um jogador clica no menu de pausa do jogo e sai para o menu inicial , o real time clock contabiliza a data atual de saída para menu, e, quando o jogador regressa ao jogo, volta a contabilizar a data de entrada no jogo , fazendo a diferença entre cada um dos tempos e subtraindo ao clock do jogador, cuja vez era de jogar.

3. Estrutura e organização de código

Timer - 5%:

O módulo do timer (timer.c) contém todas as funções que foram utilizadas para a resolução do lab2 feito ao longo do semestre, durante as aulas práticas.

Estas funções foram adaptadas para funcionarem com a nossa implementação do jogo, permitindo assim receber as interrupções do timer e alterar os contadores e frame-rate.

Teclado - 10%:

O módulo do teclado (keyboard.c) contém todas as funções que foram utilizadas para a resolução do lab3 feito ao longo do semestre, durante as aulas práticas.

Estas funções foram adaptadas para funcionarem com a nossa implementação do jogo, permitindo assim receber interrupções do teclado e validar os códigos das teclas clicadas e organizá-los de acordo com o valor recebido

Rato - 10%:

O módulo da gráfica (mouse.c) contém todas as funções que foram utilizadas para a resolução do lab4 feito ao longo do semestre, durante as aulas práticas.

Estas funções foram adaptadas para funcionarem com a nossa implementação do jogo, permitindo assim receber interrupções do rato, detectando deslocamentos do mesmo e verificar em que estado está o rato atualmente.

Gráfica - 15%:

O módulo da gráfica (graphics.c) contém todas as funções que foram utilizadas para a resolução do lab5 feito ao longo do semestre, durante as aulas práticas.

Contém também todas as funções de inicialização do modo gráfico , criação e manipulação de buffers, armazenamento e desenho de XPMs.

RTC - 5%:

O módulo do rtc (rtc.c) contém todas as funções necessárias à utilização do mesmo durante o jogo , isto é , as funções que lêem e processam a data atual, bem como funções de print que foram utilizadas no debug, ao longo do programa.

Main - 10%:

O módulo da main (main.c) contém a inicialização de todo o programa , bem como o driver_receive() loop que recebe as interrupções de cada um dos dispositivos.

Game - 20%:

O módulo do jogo (game.c) é responsável por gerenciar a lógica principal do jogo, incluindo a criação e destruição de estruturas de dados essenciais como Game, Player e Board, além de controlar o fluxo do jogo, as regras de movimentação das peças e mudança de estado do jogo.

Controller - 17%:

O módulo controller.c é o núcleo de controlo do fluxo do jogo gerenciando, mudanças de estados, e integração com outros componentes do sistema.

Também contém funções para inicialização do jogo, processamento do game loop e manipulação da informação recebida pelo teclado e pelo rato.

View - 5%:

O módulo view.c é responsável por toda a coordenação da renderização gráfica, tratando principalmente da comunicação com o módulo gráfico, invocando as funções que devem ser chamadas.

Ele comunica à gráfica onde desenha o tabuleiro, quais peças desenhar, onde e quando desenhar os relógios.

Utils - 1%:

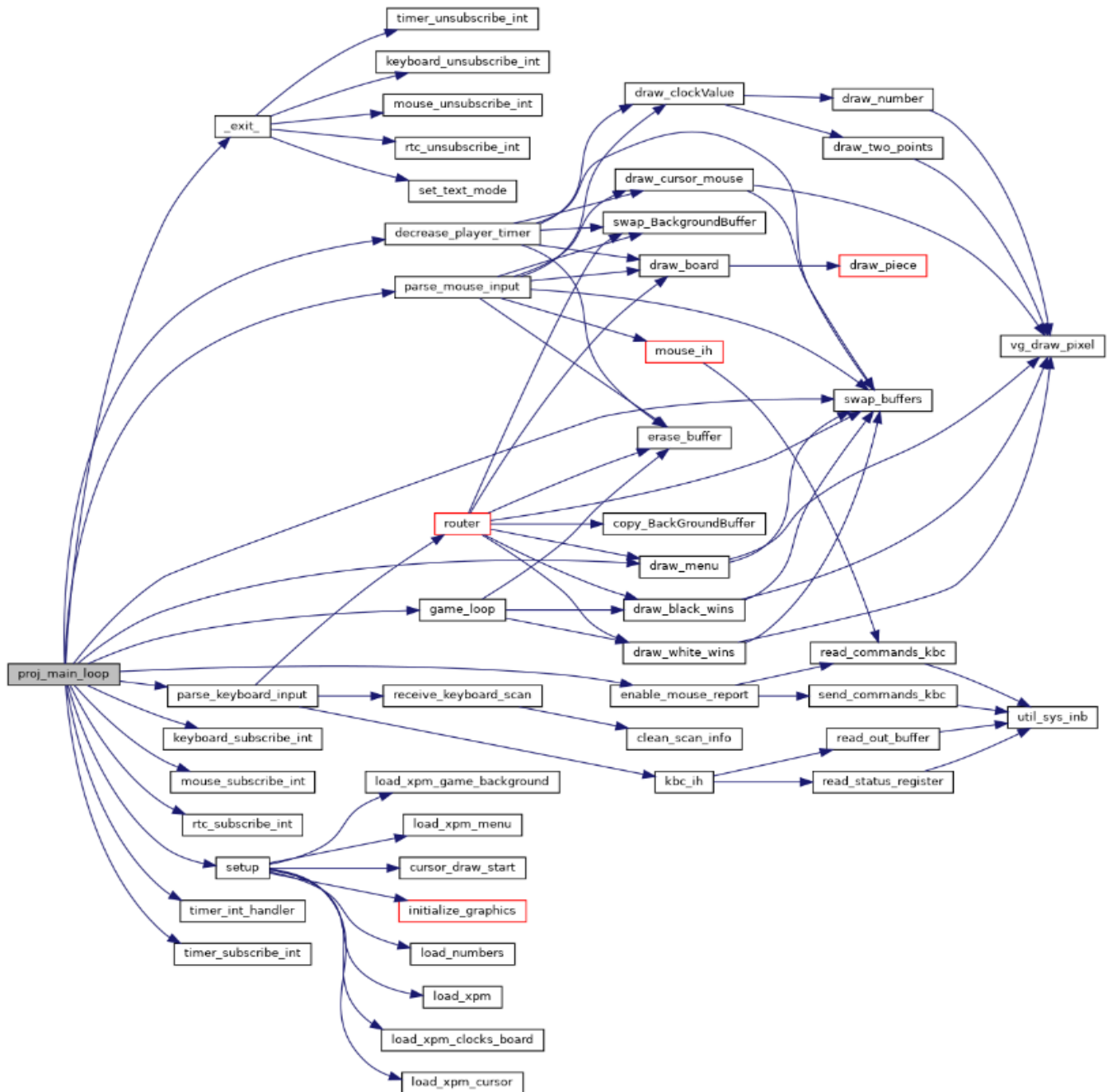
O módulo utils.c contém todas as funções que foram utilizadas para a resolução de a maioria dos labs e para obter compatibilidade com todos os módulos feitos nas aulas práticas foram assim incluídos também no projeto.

KBC - 2%:

O módulo kbc.c é responsável por mediar a comunicação entre o teclado/rato e o processador. Contém funções para ler e enviar comandos para o kbc, bem como para ler os registros de status e dados do KBC.

Tal como no utils, o seu uso foi imperioso para o lab3 e o lab4, pelo que o módulo feito nas aulas práticas foi assim incluído no projeto.

3. Gráfico de chamada de funções



Conclusão

Em jeito de conclusão, o projeto "Knights of LCOM" representa não apenas o resultado de um trabalho contínuo ao longo do semestre, mas também um mergulho profundo na minuciosidade que é toda a comunicação entre dispositivos I/O e o processador.

Através da implementação de uma interface de usuário intuitiva e responsiva, combinada com uma lógica de jogo robusta, conseguimos criar uma experiência envolvente para os jogadores. Desde a tela inicial até as telas de vitória, cada aspecto do jogo foi cuidadosamente projetado e implementado para garantir a melhor experiência possível.

Além disso, a integração de diversos dispositivos, como o teclado, rato e RTC, exigiu um planeamento adicional e o conhecimento do funcionamento de cada componente. Ao superar esses desafios, ampliamos nosso conhecimento e habilidades técnicas, preparando-nos para enfrentar futuros projetos com confiança e determinação.

De forma a continuar o projeto num futuro próximo, provavelmente as features que seriam interessantes de implementar seria um bot para jogar contra o jogador, que calculasse o melhor movimento para aquela jogada, a serial port para permitir jogar multiplayer em dois computadores distintos e alguns modos de jogos extra que exigiam extrapolar do jogo base, por exemplo, xadrez de 4 jogadores ou um modo apenas rainhas.

Com este projeto conseguimos obter bastante conhecimento de como funciona a comunicação low-level entre os dispositivos e o processador, bem como aumentar a nossa proficiência na linguagem C.