

## Connecting 4

**LCOM - 2020/2021**

Turma 2 - Grupo 12

Carlos Gomes - up201906622

Luís Miranda - up201306340

# Dispositivos Implementados

Dispositivo	Funcionalidades	Interrupções
Timer	Atualização do estado de jogo e sprites	Sim
Teclado	Controlar um dos jogadores e seleção no menu	Sim
Rato	Controlar um dos jogadores e seleção no menu	Sim
Placa Gráfica	Desenho dos diferentes sprites utilizados	Não

## Timer

A função do timer é essencialmente a atualização do estado e de jogo e da sua representação e possibilita a animação de sprites.

Funções:

- update\_game(Game \*game);
- void piece\_animation(Game \*game, Disc \*d);
- subscribe\_interruptions(game);
- void exit\_game(Game\* game);
- funções do ficheiro timer.c

## Teclado

O teclado é utilizado para escolher e efetuar a jogada por um dos jogadores. Para isto são utilizadas as setas e a tecla “Enter”. Também é possível voltar do jogo para o menu inicial ou sair completamente do jogo ao premir a tecla “ESC”.

Funções:

- update\_game(Game \*game);
- subscribe\_interruptions(game);
- Game \*initiate\_game();

- kbd\_game\_handler(Game \*game);
- void exit\_game(Game\* game);
- funções do ficheiro keyboard.c

## **Rato**

Tal como o teclado, o rato permite percorrer e seleccionar a opção pretendida no menu principal, e durante o jogo permite efetuar a jogada do jogador adversário. Para tal é registado o movimento e posição do rato bem como o premir do botão esquerdo tanto nos menus como durante o jogo.

Funções:

- update\_game(Game \*game);
- subscribe\_interruptions(game);
- Game \*initiate\_game();
- mouse\_game\_handler(Game\* game);
- mouse\_follow\_disc(Game \* game);
- void exit\_game(Game\* game);
- funções do ficheiro mouse.c

## **Placa Gráfica**

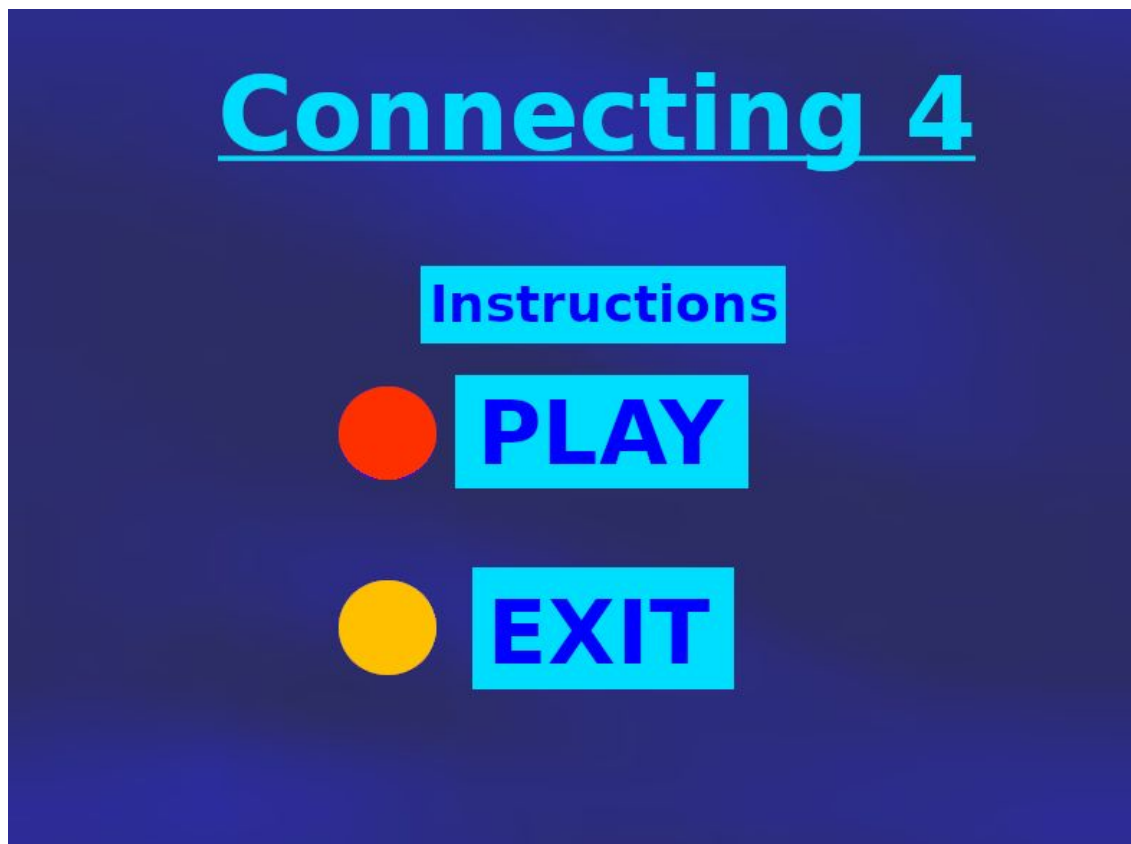
A placa gráfica é responsável por toda a representação e desenho das várias imagens e sprites do jogo, desde o menu, ao tabuleiro, aos discos, etc. O modo que decidimos utilizar foi o 0x115, de resolução 800x600. Foi utilizado double buffering e implementadas animações dos sprites das peças para as jogadas.

Funções:

- init\_graphics\_mode();
- Game \*initiate\_game();
- void exit\_game(Game\* game);
- funções do ficheiro graphics\_mode.c

# Instruções de Utilização

## Menu Principal

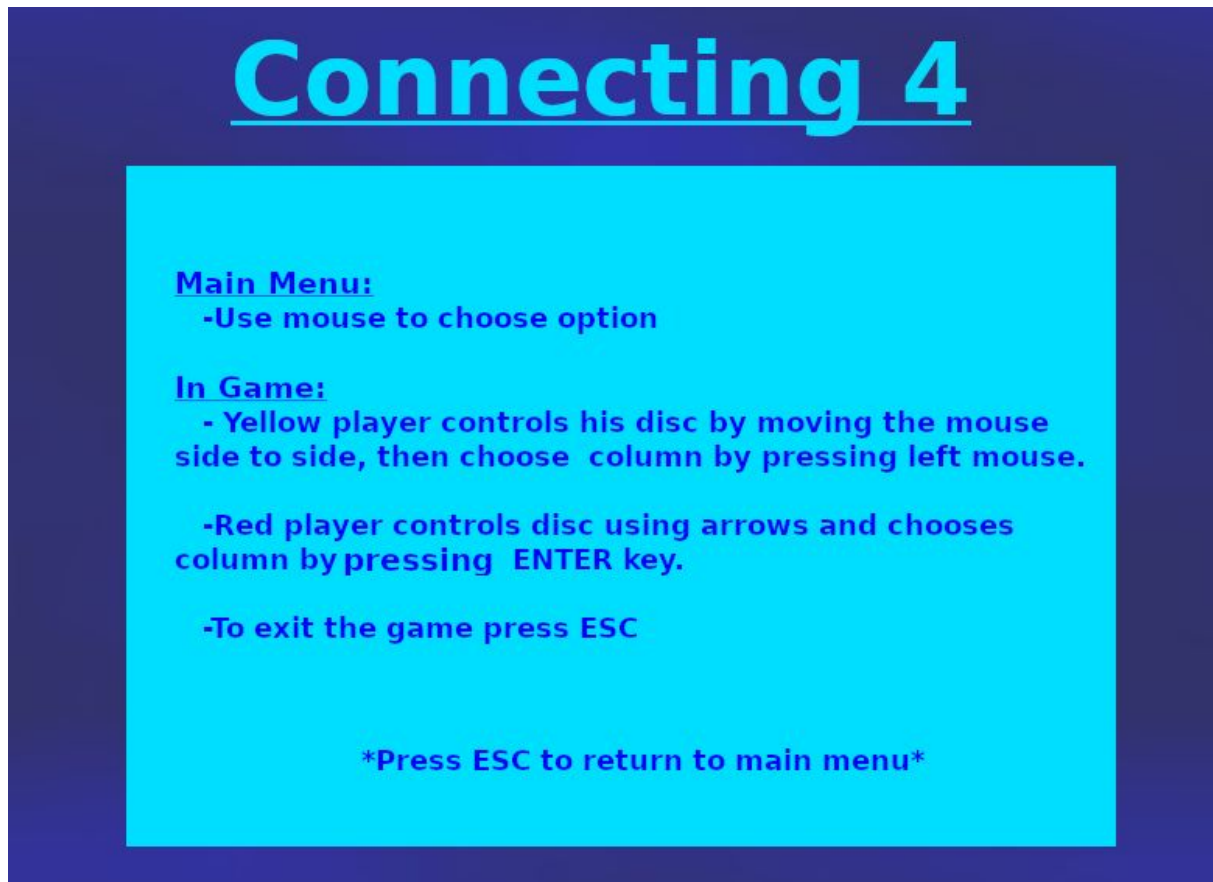


Quando o jogo é iniciado, ao utilizador é apresentado o menu inicial, que permite a escolha de uma de três opções:

- Instruções
- Jogar
- Sair

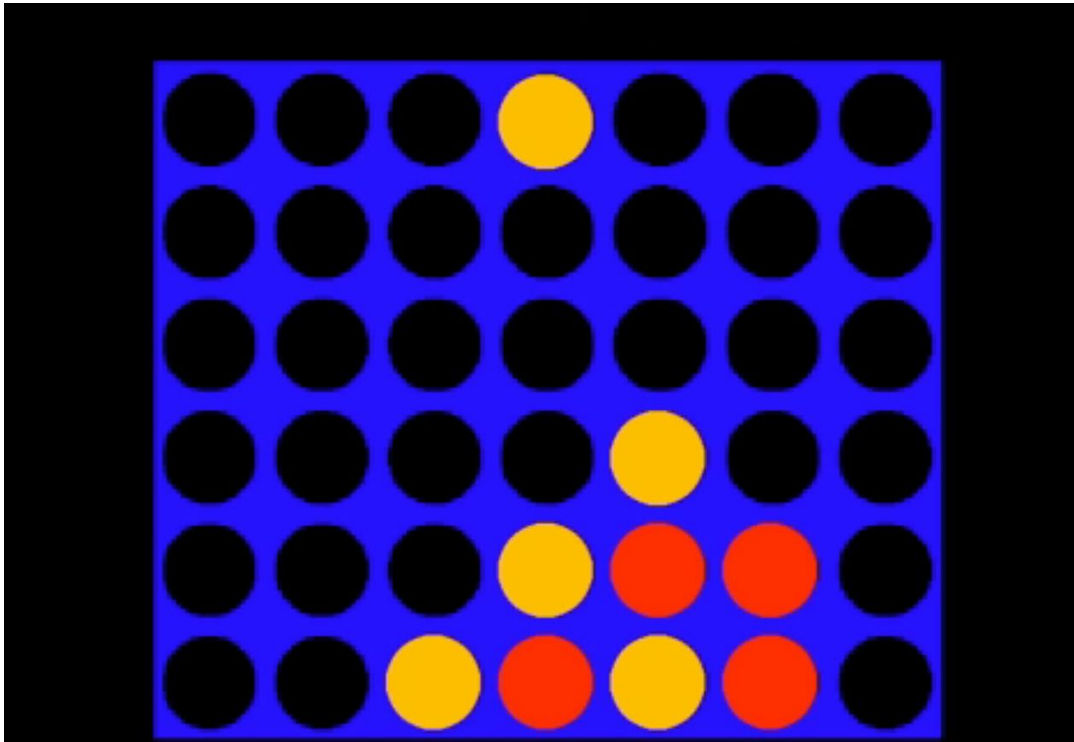
A escolha destas opções pode ser feita com o rato, utilizando a sua posição e botão esquerdo para efetuar a seleção.

## Instruções



Mostra uma tela com os controlos de jogo, tanto para o menu como para o jogo em si.

## Jogar



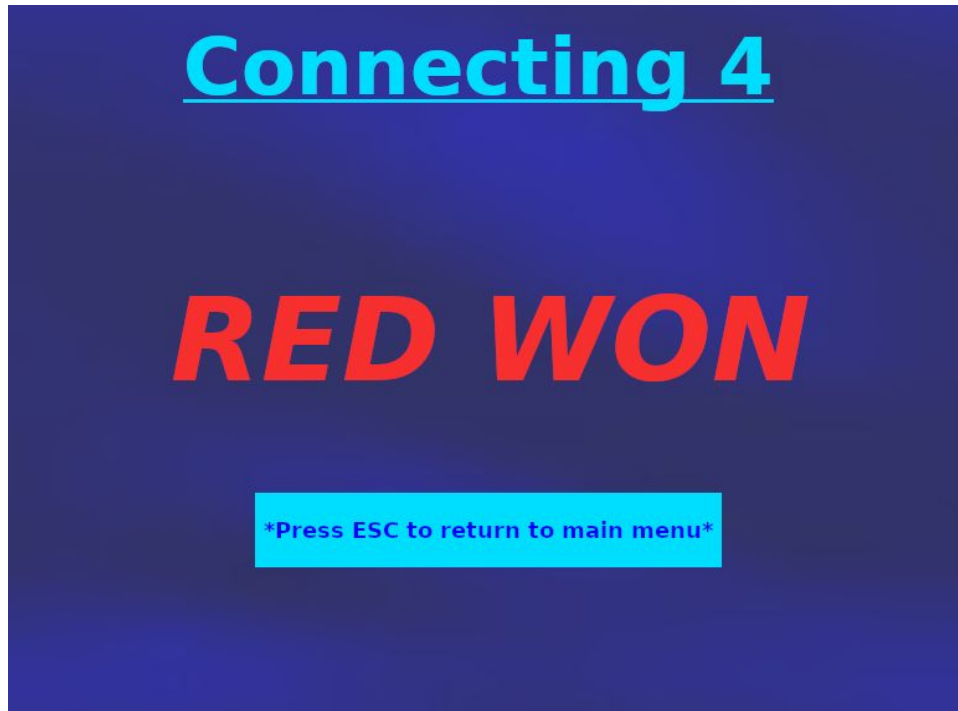
Os jogadores vão colocando peças alternadamente com o objetivo de conseguir colocar 4 peças em sequência (na horizontal, vertical ou diagonal). No caso do tabuleiro ser preenchido sem que nenhum dos jogadores o consiga, é declarado um empate.

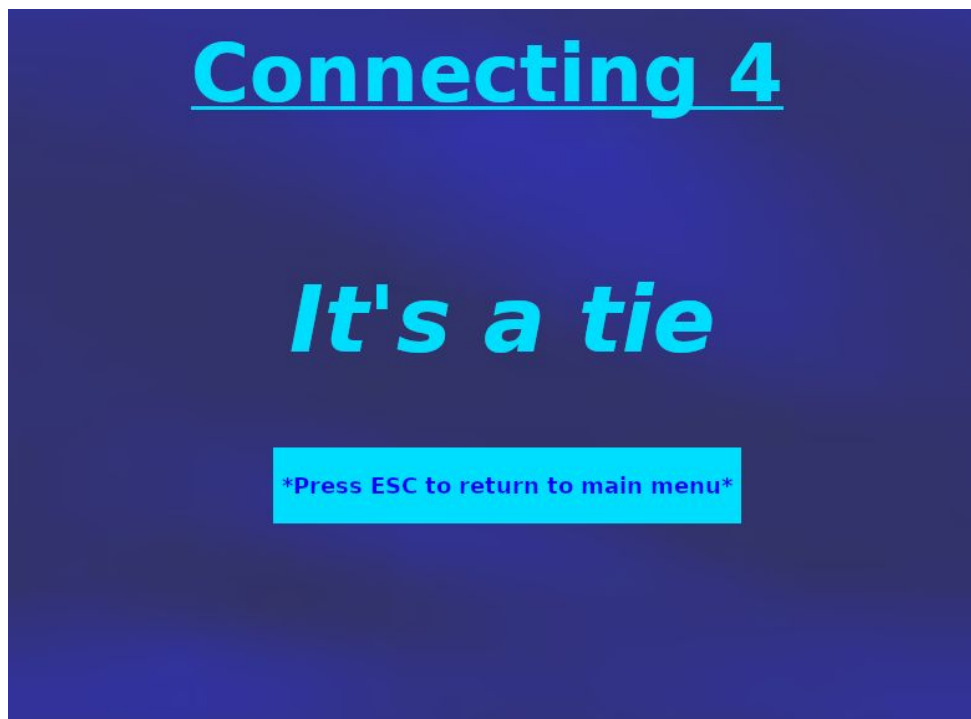
Cada jogador controla uma das peças. O “Jogador 1” controla o disco amarelo, movendo-o com o movimento do rato e selecionando a coluna desejada com o botão esquerdo do rato. O “Jogador 2” move as peças vermelhas com a ajuda das setas (esquerda e direita) e seleciona a coluna desejada ao pressionar a tecla “Enter”. A qualquer altura o jogo pode ser terminado e voltar ao menu anterior pressionando a tecla “ESC”.

## Sair

Termina o programa, sendo o utilizador levado de volta para o terminal do Minix.

## Game Over Screen





Quando um dos jogadores vence o jogo (conseguindo uma sequência de 4 peças) ou caso o jogo termine empatado, é apresentada uma das anteriores telas (consoante o desfecho do jogo). É mostrada a informação de como voltar ao menu inicial (premir a tecla ESC) para o jogador ser levado de volta para escolher novamente uma das opções disponíveis.

## Organização do Código

### **game.c**

Neste ficheiro estão implementadas as funções relativas a interrupções e toda a lógica do jogo, desde a execução de uma jogada, a deteção de um vencedor, etc.

Peso do módulo no projeto: 40%

Módulo desenvolvido por: Carlos Gomes e Luís Miranda (50%/50%)

### **graphics\_mode.c**



Responsável pela representação de todas as imagens do projeto. Importado do “Lab 5”, com ligeiras modificações.

Peso do módulo no projeto: 10%  
Módulo desenvolvido por: Carlos Gomes

## **keyboard.c**

Responsável pela interação com o teclado. Importado do “Lab 3” com ligeiras modificações.

Peso do módulo no projeto: 10%  
Módulo desenvolvido por: Luís Miranda

## **main\_menu.c**

Representação do menu inicial, e controlo da seleção da opção desejada.

Peso do módulo no projeto: 5%  
Módulo desenvolvido por: Carlos Gomes

## **mouse.c**

Neste ficheiro encontram-se as funções responsáveis pela leitura e manipulação dos pacotes do rato. Importado do “Lab 4” embora com algumas alterações.

Peso do módulo no projeto: 15%  
Módulo desenvolvido por: Luís Miranda

## **proj.c**

Ficheiro responsável pelo “main loop” do programa, com iniciação do modo gráfico e consequentemente do jogo em si.

Peso do módulo no projeto: 2%  
Módulo desenvolvido por: Carlos Gomes e Luís Miranda (50% / 50%)

## **sprite.c**

Ficheiro que inclui as funções que permitem a criação e representação das diferentes sprites necessárias para o jogo (tabuleiro, discos, background). Também inclui funções para a movimentação dos discos.

Peso do módulo no projeto: 10%

Módulo desenvolvido por: Carlos Gomes

## **timer.c**

Responsável pela interação do programa com o timer. Importado do “Lab 2”, com ligeiras alterações.

Peso do módulo no projeto: 5%

Módulo desenvolvido por: Carlos Gomes e Luís Miranda (50%/50%)

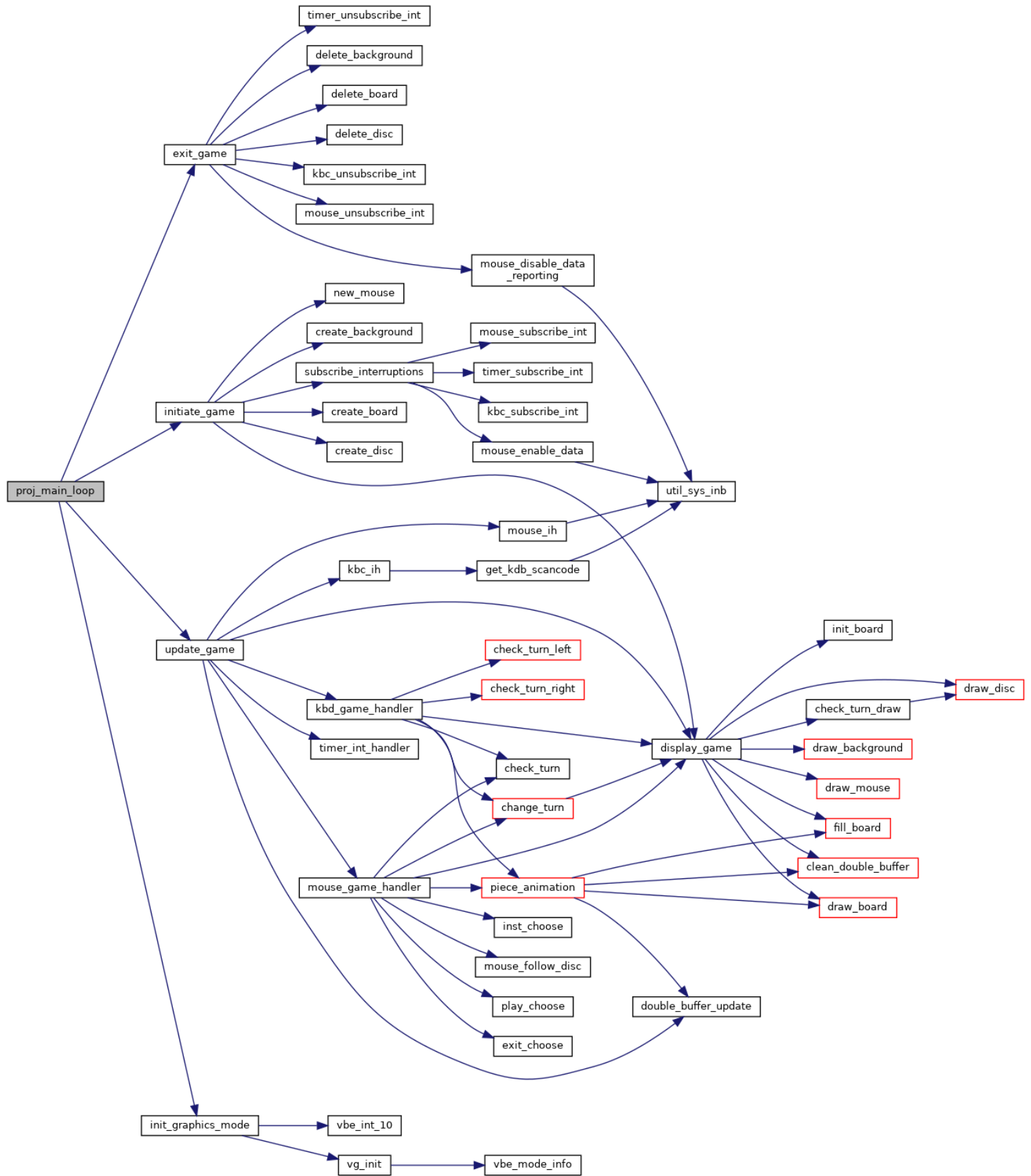
## **utils.c**

Importado do “Lab 2” permite a utilização de certas funções de ajuda.

Peso do módulo no projeto: 3%

Módulo desenvolvido por: Carlos Gomes e Luís Miranda (50%/50%)

# Gráfico de Chamadas de Função



## **Detalhes de Implementação**

A implementação dos diferentes dispositivos foi bastante standart e semelhante ao que já tinha sido feito nos Labs respetivos. No entanto, aquando da implementação do método de double buffer, foi possível verificar em certas máquinas (maioritariamente em windows), um delay no processamento dos packets do rato. É possível obter certas melhorias neste campo alterando a velocidade com que o double buffer é atualizado, no entanto estes resultados verificam-se bastante dependentes do computador e sistema operativos em que a máquina virtual esta a ser executada.

## **Conclusão**

Pensamos que o projeto foi concluído com sucesso, sendo que conseguimos implementar todas as funcionalidades propostas, no entanto é possível em certos computadores que se verifique um pouco de lag do rato devido à implementação do double buffer, algo que pensamos ser um problema do Minix.

Quanto à UC pensamos que as aulas contribuíram positivamente para a obtenção dos conhecimentos necessários para desenvolver o projeto, no entanto perde-se bastante ao nível de produtividade ao trabalhar com um colega à distância, sendo que os grupos poderiam ser formados por elementos do mesmo turno para melhorar este aspeto.