

LCOM PROJECT 2021/2022

LCOMEMORY

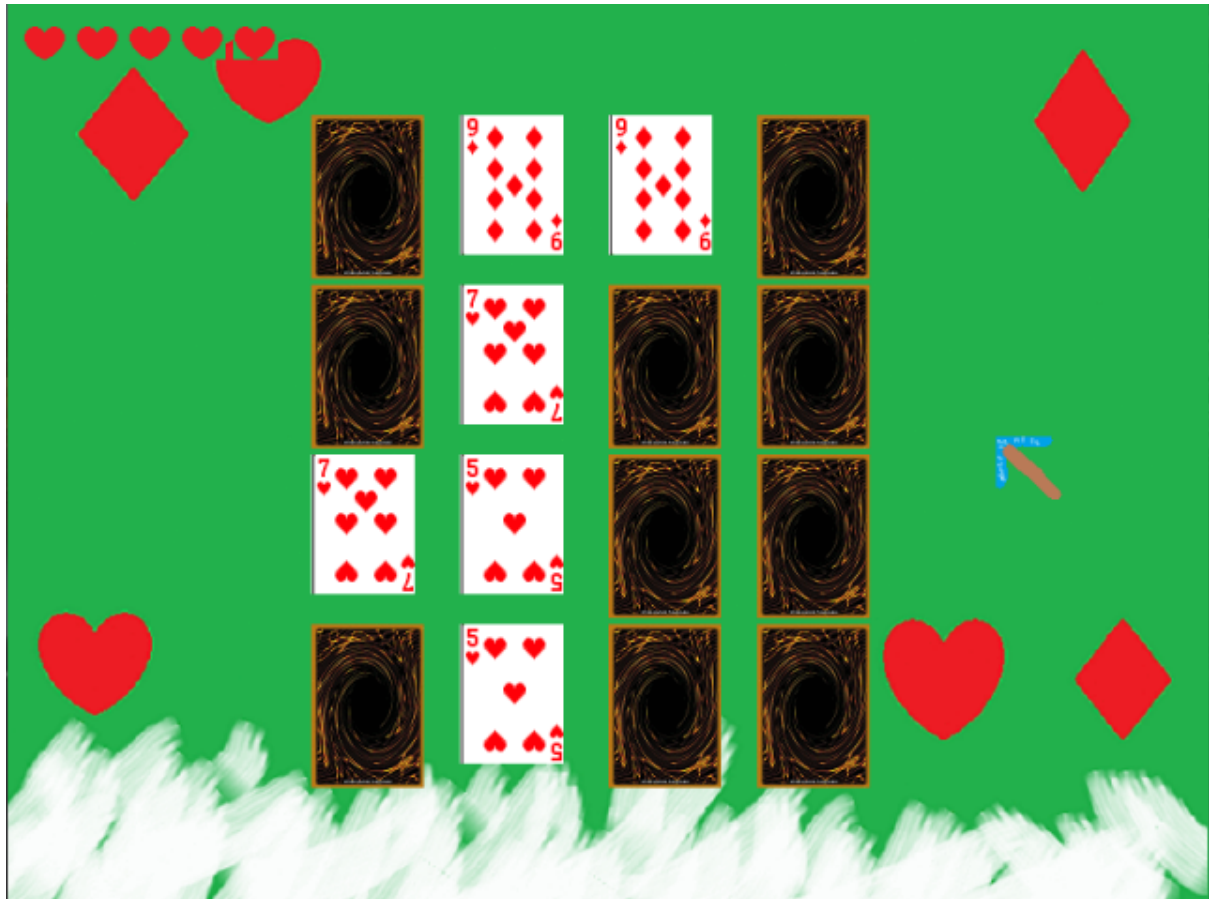


Figura 1

Realizado pelos estudantes:

- | | | |
|----------------|---|-------------|
| ❖ André Costa | - | up201905916 |
| ❖ Filipe Pinto | - | up201907747 |
| ❖ João Rodrigo | - | up201705110 |

Índice

1. Introdução	2
2. Instruções de utilização	2
3. Periféricos Implementados (Estado do projeto)	9
4. Estrutura e organização do código	12
5. Detalhes de implementação	18
6. Conclusão	19

Introdução

LCOMEMORY, tal como o nome diz, é um jogo de memória, cujo objetivo é, dentro de um conjunto de cartas dado, selecionar aos pares as mesmas, terminando o jogo com todas as cartas viradas para cima, sem exceder o número de falhas (vidas do jogador).

Instruções de utilização

❖ Menu

Ao iniciar o jogo, aparecerá o Menu. Este é constituído por um background, 3 botões e uma imagem referente à hora do dia (indicando se é de noite ou se é de dia), como podemos observar na figura 2.

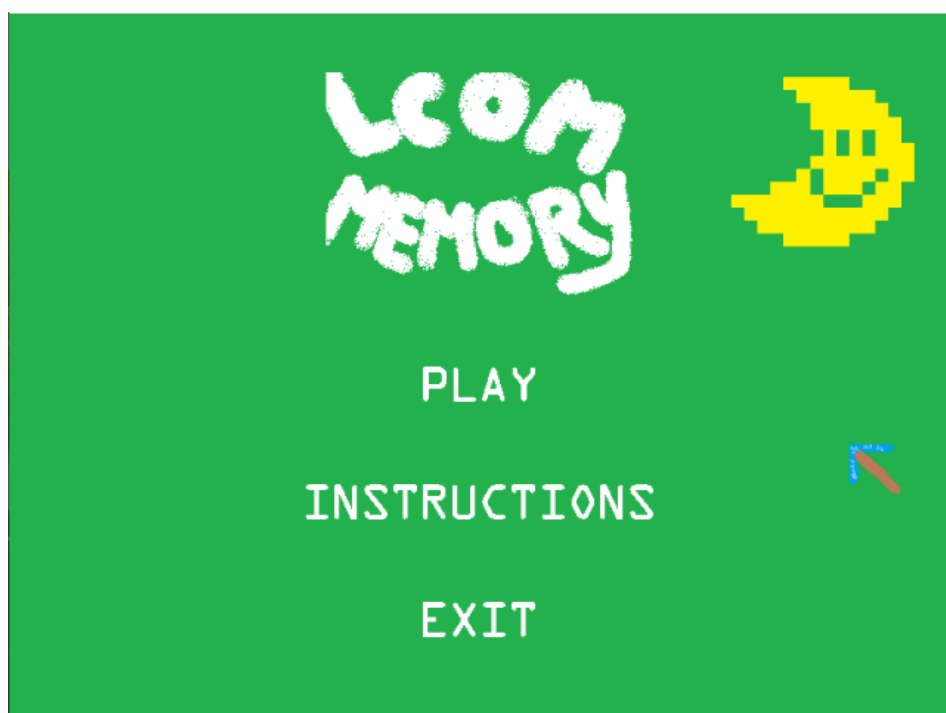


Figura 2

❖ Game

Sendo o primeiro botão (Play) selecionado, o jogo será iniciado. Mal o jogo começa, as cartas serão mostradas ao jogador durante 2 segundos para que este consiga decorar as suas posições no tabuleiro. Após o tempo passar, as cartas serão viradas para baixo instantaneamente, dando liberdade ao jogador de começar as jogadas.

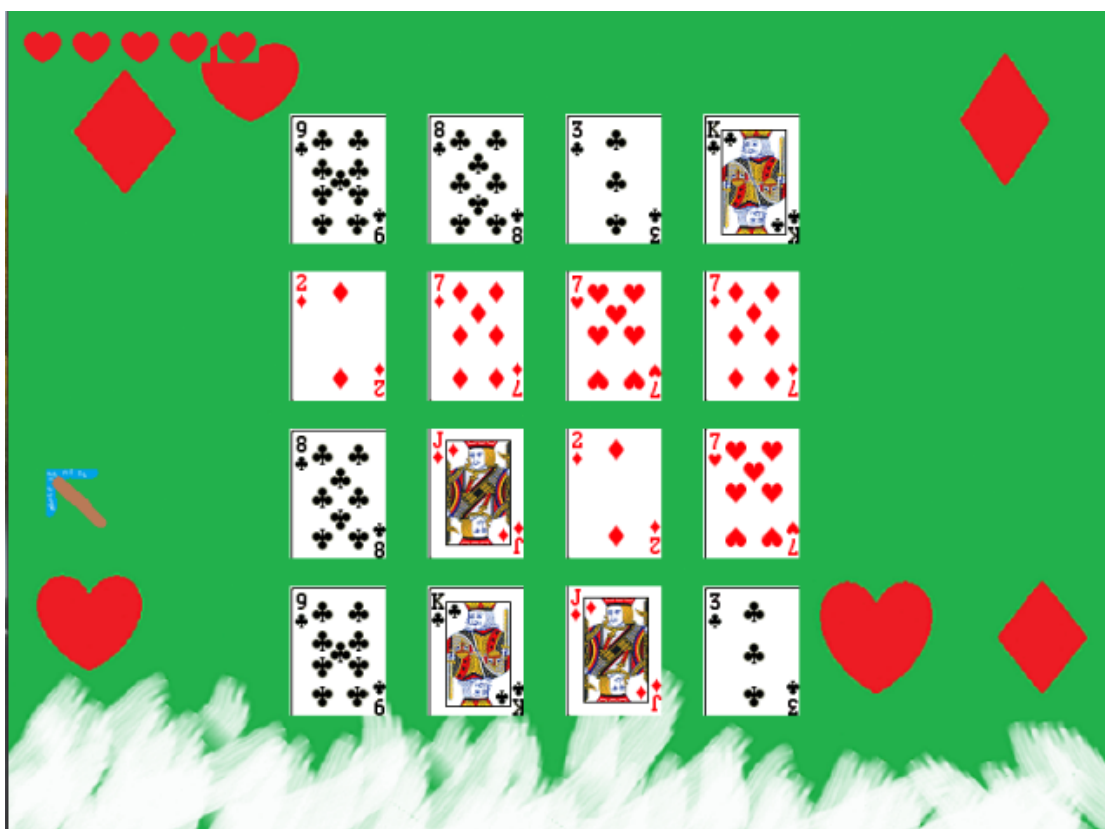


Figura 3



Figura 4

➤ Pause

Carregando na tecla "ESC", o menu de Pausa será mostrado. Este é constituído por 3 botões, tal como o Menu principal. A diferença é que em vez do botão Play, tem um para dar continuação ao jogo pausado (Resume).

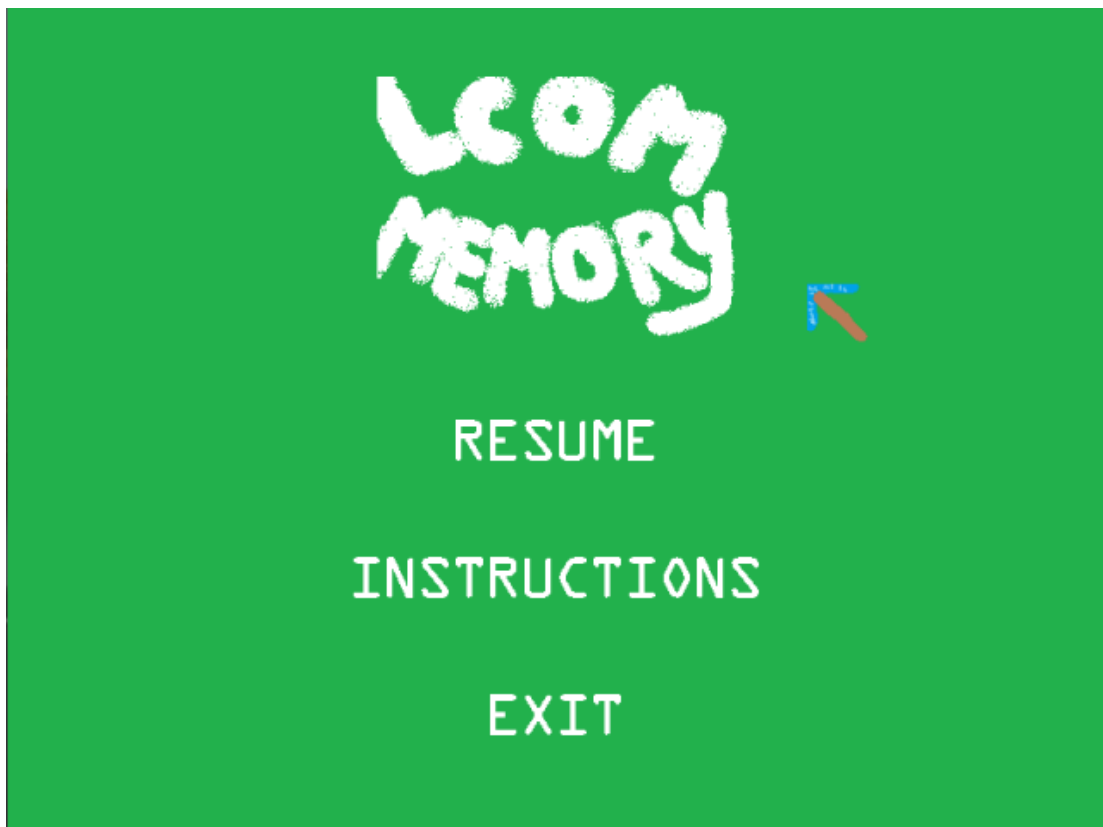


Figura 5

➤ Vitória

Concluindo o objetivo do jogo sem perder as 3 vidas do jogador, o ecrã de Vitória será mostrado e voltará para o Menu principal após 2 segundos.



Figura 6

➤ Derrota

Fazendo jogadas erradas, ou seja, escolhendo 2 cartas que não completem um par, o jogador perderá 1 vida. Ficando sem vidas, o jogo chegará a um fim, mostrando o ecrã da Derrota (Game Over) e voltará para o Menu principal após 2 segundos.

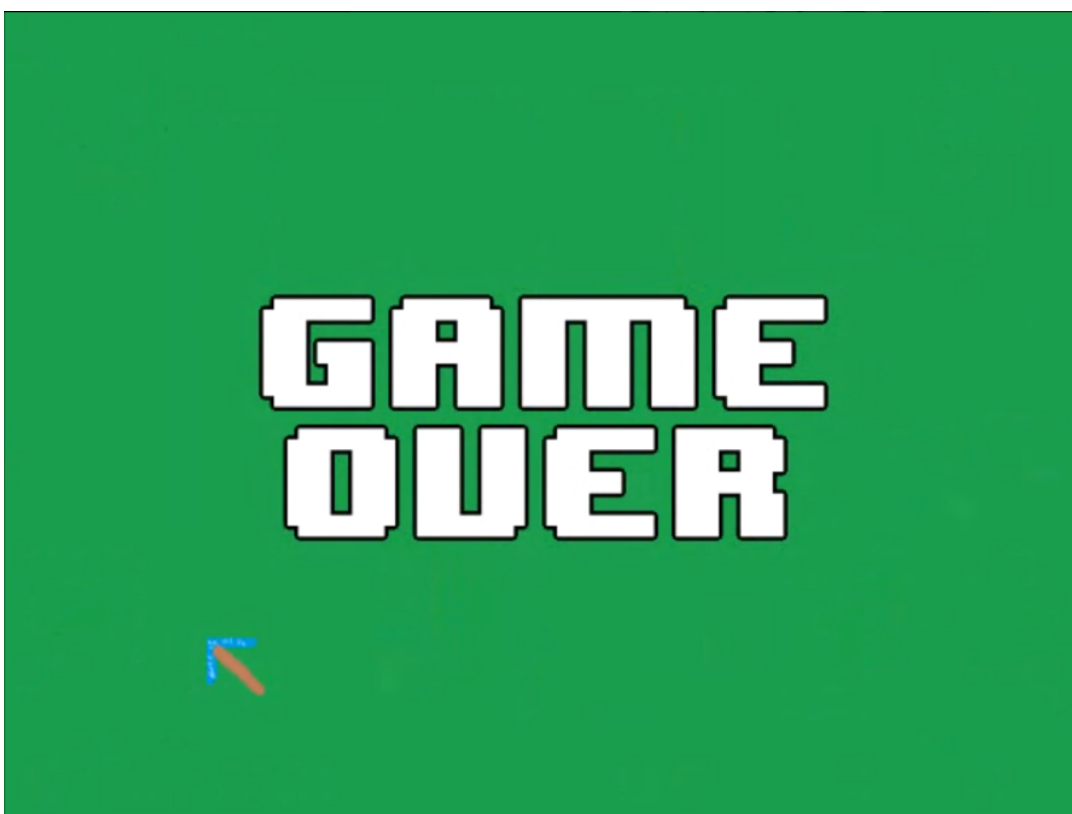


Figura 7

❖ Instruções

Sendo o segundo botão selecionado (Instructions), aparecerão as Instruções do jogo. O utilizador deverá lê-las atentamente, antes de começar a jogar.



Figura 8

❖ Exit Game

Sendo o terceiro botão selecionado (Exit), o jogo será encerrado.

Periféricos Implementados (Estado do Projeto)

A tabela a seguir representada contém a informação resumida de que periféricos implementamos, de que forma os usamos e se utilizamos ou não interrupções.

Periférico	Utilização	Interrupções
Timer	Controlar a frame rate	Sim
Keyboard	ESC - abrir o Pause P - para iniciar o jogo I - mostrar as intruções	Sim
Video Card	Mostrar os Menus e o jogo	Não
Mouse	Selecionar os botões do menu. Selecionar as cartas	Sim
RTC	Mediante a hora e data atual, é apresentado uma imagem a diferenciar o dia e a noite	Não
Serial Port	-	Não

Timer:

O periférico timer, foi utilizado para controlar a frame rate do jogo.

Para esse controle, utilizamos as interrupções do timer com uma frequência de 60 interrupções por segundo.

Usamos as funções, implementadas já no Lab2,

timer_subscribe_int, *timer_unsubscribe_int* e *timer_int_handler*.

Keyboard:

O periférico keyboard foi utilizado para abrir o Menu de Pause através da tecla "Esc", para começar o jogo através da tecla "P" e carregando na tecla "I" para abrir as Instruções.

Para isso, utilizamos as interrupções do keyboard. Usamos as funções, implementadas no Lab3, *kbc_subscribe*, *kbc_unsubscribe* e *kbc_ih*.

Video Card:

O periférico da video-card foi utilizado para dar display dos menus e do jogo. O modo utilizado foi o 0x115 (800x600 de resolução) com 24 bits per pixel, o que resulta num conjunto de 2^{24} cores disponíveis, usando RGB 8-8-8 (8 pixels para cada cor).

As funções utilizadas, implementadas no Lab5, foram *vg_start* responsável por trocar o modo do vídeo, *my_updateVideoMem* onde usamos *double buffering*, *my_draw_pixel* e outras 2 novas funções que implementamos. A função *greenScreen* que mete o ecrã todo a verde e a função *vg_draw_card* que desenha uma carta numa certa posição.

Mouse:

O periférico do mouse foi utilizado tanto nos menus para selecionar os botões, como durante o jogo para selecionar as cartas (usando o botão do lado esquerdo do rato).

Funciona através de interrupções, por isso sempre que a posição é mudada ou uma tecla é pressionada, é gerada uma interrupção que é tratada pelo *mouse_ih*.

As funções utilizadas, implementadas no Lab4, foram *mouse_subscribe*, *mouse_enable_data_reporting*, *disable_mouse_data* e *mouse_unsubscribe*.

RTC:

O periférico do rtc foi utilizado para preencher uma struct que criámos que armazena a data e hora atual. Mediante a hora do dia, desenhámos uma imagem que diferencia, no jogo, se está de dia ou de noite, tendo em conta a hora atual.

As funções implementadas neste periférico, foram *rtc_subscribe*, *rtc_unsubscribe*, *convert_bcd_to_dec* e *get_date*.

Estrutura e organização do código

No desenvolvimento do projeto, fomos dividindo o código em diversos módulos, para uma melhor compreensão e legibilidade. Os diagramas de chamadas a funções foram gerados com o Doxygen, pelo que, no relatório apenas estão presentes os mais importantes.

Módulos relacionados com os Periféricos

Os módulos apresentados a seguir referem-se ao manuseamento dos periféricos, cujos foram desenvolvidos ao longo das aulas práticas (os Labs). Estes módulos foram feitos por todos os elementos do grupo.

Módulo do Timer "timer":

Módulo desenvolvido no Lab2, cujo contém as funções referentes ao handler e à subscrição das interrupções do timer.

Módulo i8254:

Módulo também criado durante o Lab2, cujo contém as constantes referentes ao timer.

Módulo do Keyboard "keyboard":

Módulo desenvolvido no Lab3. Neste módulo encontram-se as funções referentes à manipulação do teclado (subscrições e handler das interrupções).

Módulo i8042:

Módulo desenvolvido para o Lab3. Este módulo contém as constantes utilizadas pelo keyboard.

Módulo do Mouse "mouse":

Módulo desenvolvido durante o projeto, cujo contém as funções referentes ao handler e à subscrição das interrupções do mouse.

Módulo da Video Card "video":

Módulo desenvolvido no Lab5 e durante o projeto. Neste módulo para além das funções desenvolvidas no lab para desenhar um pixel, inicializar o modo, copiar do buffer para a memória, também acrescentamos funções referentes ao projeto, como desenhar uma carta, meter o ecrã todo a verde.

Módulo do RTC:

Módulo desenvolvido durante o projeto. Neste módulo desenvolvemos funções que obtêm a data e hora atual do computador recorrendo ao Real-Time Clock do computador.

Neste módulo preenchemos a struct que armazena a data e hora atual lendo e escrevendo para os devidos registos.

Módulo pixmap:

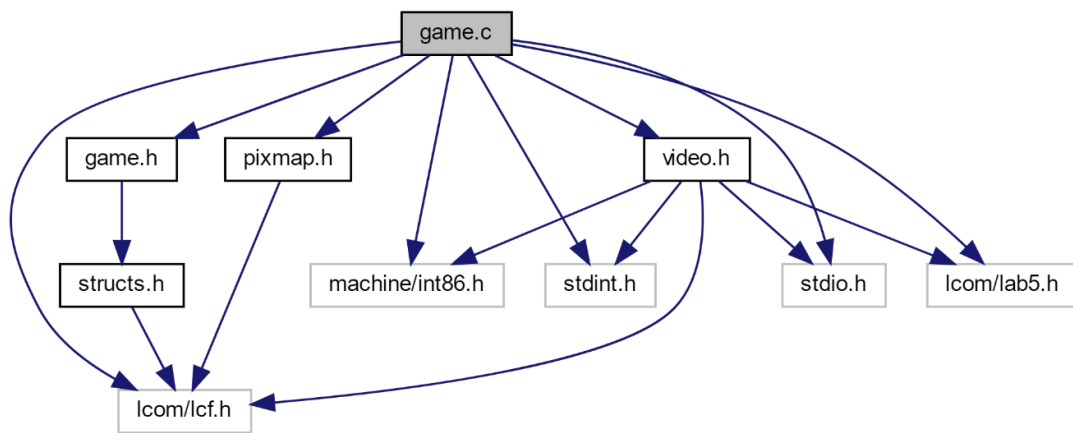
Este módulo contém todas as imagens que utilizamos no projeto, convertidas em xpm's, tal como todo o alfabeto, para construção de palavras, como os botões.

Este módulo foi implementado por todos os membros à medida que íamos precisando de novas imagens para objetos.

Módulo do Jogo "game"

Este modo é responsável por controlar todos os objetos do jogo. Desde o jogador e o conjunto de cartas (vetor) até aos periféricos, o módulo game controla as chamadas às funções de cada módulo.

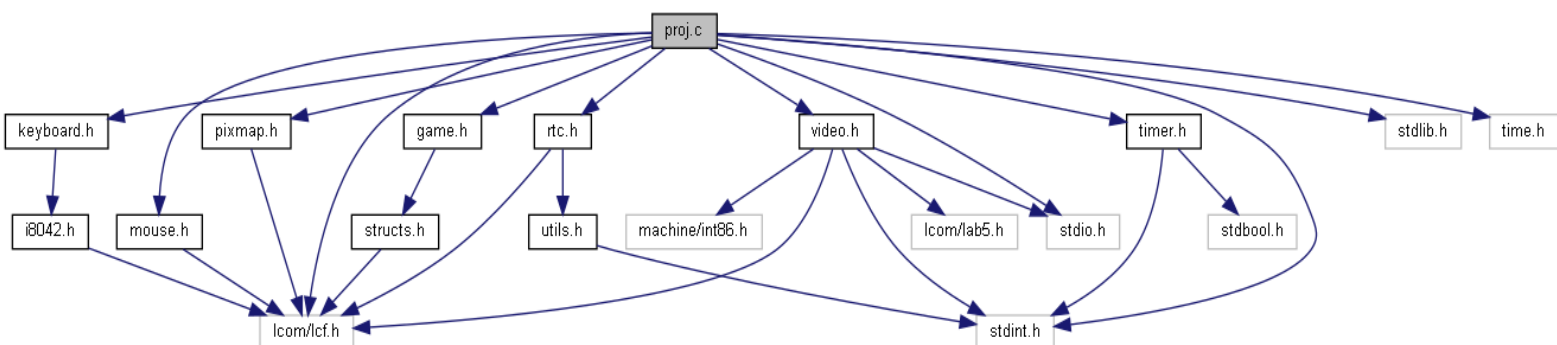
É neste módulo que desenhamos todos os componentes do nosso jogo, que atualizamos as suas posições e o seu estado.

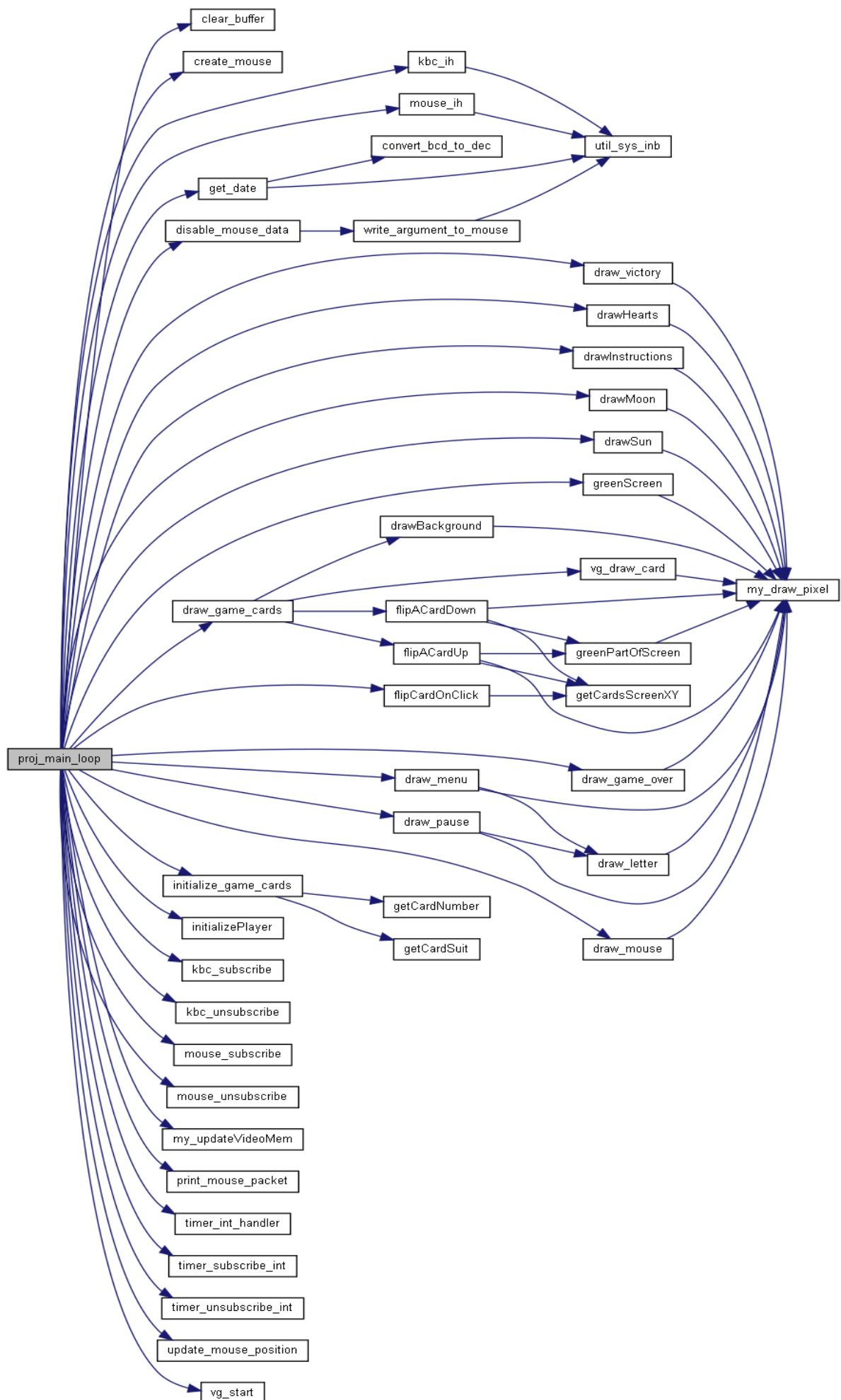


Módulo Projeto "proj":

Módulo principal/inicial do nosso projeto. A função `main` está presente neste módulo. Este módulo é o responsável por controlar o ciclo das interrupções de todos os periféricos (chamada à função `driver_receive()` e onde são chamados os interrupt handlers).

Também é responsável por dar load dos xpm's declarados no módulo "pixmap".





Tanto este módulo como o anterior (game) foram implementados por todos os membros do grupo, tendo mais percentagem o João Rodrigo.

Documentação do Projeto

Todo projeto foi devidamente documentado utilizando o Doxygen. Também foi selecionada a opção de gerar diagramas de chamadas a funções, já anteriormente apresentados.

Detalhes da Implementação

No decorrer do projeto, tal como nos Labs, implementamos as funções de forma ao código ficar bem dividido e organizado em módulos, de modo a facilitar a compreensão como também a estruturação do projeto.

Utilizamos também uma **máquina de estados** que indicava em que ecrã/state se situava o jogador. Isto é, o jogador começaria o jogo sempre no state Menu, onde poderia escolher o próximo state do jogo, perante o botão que clicasse com o rato ou perante a tecla do teclado que fosse carregada.

O nosso jogo é todo **orientado a objetos**, sendo todos os elementos do nosso jogo objetos de uma classe. Estas classes estão definidas no ficheiro "structs.h" e os objetos são todos definidos no módulo Game.

Em relação à **geração de frames**, todos os eventos do jogo são gerados quando ocorrem interrupções, sendo posteriormente "tratados". A frequência utilizada no timer foi 60 Hz, ou seja, a cada 60 ticks do timer por segundo todos os objetos são desenhados num buffer e posteriormente copiados para o buffer da memória do video, estratégia do *double buffering*, através da função `my_updateVideoMem`.

Conclusão

A cadeira de LCOM, foi sem dúvida uma experiência diferente e interessante, fazendo referência à interação com os periféricos.

Foi também uma das cadeiras mais difíceis e desafiantes que até agora experienciamos. O trabalho autônomo e contínuo por parte de cada um foi crucial e extremamente necessário para a realização da cadeira.

Devido ao encurtamento do semestre, sentimos que a matéria sobre a porta série e sobre o rtc não foi aprofundada como gostaríamos que fosse. Acabamos por decidir fazer um jogo single player, no entanto se não tivesse sido essa a escolha, a porta série seria um periférico crucial na implementação.