



Memory Match Mayhem

Laboratório de Computadores

Grupo 4 – Turma 04

Licenciatura em Engenharia Informática e Computação

Diogo Azevedo Lemos - up202003484

Eduardo Machado Teixeira de Sousa - up202103342

João Brandão Alves - up202108670

João Pedro Sá Torres Neiva Passos - up202108833

Índice

1. Introdução	3
2. Jogabilidade	4
2.1 Main Menu	4
2.2 Game Menu	5
2.3. Menu Final	6
3. Estado do Projeto	7
3.1. Lista de Dispositivos Usados	7
3.2. Timer	7
3.3. Keyboard	7
3.4. Mouse	8
3.5. Vídeo	8
4. Estrutura do Código	9
4.1. Timer	9
4.2 Keyboard	10
4.4 Graphics	10
4.5 Model	10
4.6 View	10
4.7 Sprites	11
4.8 Utils	11
5. Detalhes de implementação	12
5.1. Máquina de Estados	12
5.2. Modelo MVC	12
5.3. Animações	13
6. Gráficos de Chamadas de Funções	14
7. Conclusões	16

1. Introdução

Como projeto final para a cadeira de Laboratório de Computadores, o nosso grupo decidiu explorar a possibilidade de criar um jogo que fosse em retorno de outros tempos, um jogo intemporal. O jogo da memória “*Memory Match Mayhem*”.

Como tema optamos pelo mundo jurássico visto que o aspeto visual é extremamente importante num jogo da memória. No projeto usamos timer, keyboard, mouse e video.

2. Jogabilidade

O objetivo deste jogo é conseguir completar todos os pares de cartas de dinossauros disponíveis. O jogo começa com todas as cartas viradas de face para baixo de modo a que o jogador tenha que virar duas cartas de forma a conseguir juntar um par tirando-o do jogo e ficando cada vez mais perto de vencer.

O objetivo é memorizar o maior número de cartas de modo a que seja mais fácil a identificação dos pares. No Memory Match Mayhem temos três modos de dificuldade, sendo que aumenta o número de cartas dispostas e de pares a serem completos. O modo fácil consiste numa disposição de 8 cartas num 4 por 2, o modo médio consiste numa disposição 4 por 3 e o modo difícil consiste então em 16 cartas dispostas em 4 por 4.

2.1 Main Menu

O menu principal consiste num sistema intuitivo que dá ao utilizador 4 escolhas. Jogar no modo fácil, médio ou difícil, e a opção de sair do jogo. O cursor indica ao utilizador quando há uma possibilidade de clique quando este muda para um dinossauro diferente.





cursor normal (esquerda)

cursor de hover (direita)

Dois tipos de cursores

2.2 Game Menu

No modo fácil como dito anteriormente é apresentado ao utilizador um tabuleiro de 4 por 2 cartas consistindo em 4 pares de dinossauros diferentes. O modo médio tem assim como tabuleiro um 4 por 3 com 6 pares de dinossauros e o modo difícil tem o maior dos tabuleiros com o tamanho 4 por 4 com 8 pares de dinossauros. Para todos os modos os dinossauros quando são emparelhados corretamente ficam voltados de face para cima indicando que apenas as cartas de face para baixo faltam para completar o jogo.

Para jogar, o utilizador pode clicar na carta que pretende virar ou usar as teclas WASD para se movimentar observando o movimento da moldura à volta das cartas indicando a carta onde se encontra. Finalmente para virar basta carregar o ENTER. Usando o botão do meio ou direito do rato leva o jogador de volta ao menu inicial.



modo fácil

modo médio



modo difícil

2.3. Menu Final

Após completar o jogo no modo escolhido, o jogador é levado a um ecrã de mensagem de parabéns em que tem duas escolhas, jogar de novo ou voltar ao menu inicial. Jogar de novo leva para um jogo da mesma dificuldade que o anterior. Ambas podem ser escolhidas com rato ou teclado.



menu final

3. Estado do Projeto

3.1. Lista de Dispositivos Usados

Dispositivo	Funcionalidade	Uso de Interrupções
Timer	Controlo do framerate, atualização do estado do jogo, double buffering temporizado e um timer simples no jogo	Sim
Keyboard	Navegação nos menus e nos jogos	Sim
Mouse	Navegação nos menus e cliques durante o jogo	Sim
Graphics	Desenho da interface do jogo	Sim

3.2. Timer

Usamos o timer para controlar o frame rate do jogo sendo este 60 frames por segundo. O timer é assim um dispositivo bastante importante na integridade do projeto pois este influencia todos os outros devido às interrupções. O double buffering também faz uso deste, sendo que a cada interrupção gerada pelo timer a memória do buffer principal é copiada para um buffer secundário. O timer também é usado durante o jogo, mostrando o tempo que o jogador está a demorar a resolver o novo jogo e para controlar o tempo das cartas voltarem a virar quando o utilizador não acerta num par.

3.3. Keyboard

Este dispositivo é utilizado na lógica do jogo e em todos os menus existentes. Todas as decisões no jogo podem ser controladas pelo teclado, visto que nos menus temos a possibilidade de selecionar o modo que o jogador pretender através de certas letras que são especificadas entre parênteses. Por

exemplo: Hard (H), Medium (M) ou Easy (E) ao pressionar a tecla H, M ou E, respetivamente, o jogador é assim apresentado o nível com a dificuldade escolhida.

As teclas WASD (Cima, Esquerda, Baixo, Direita, respetivamente) e ENTER (seleção) são as usadas para interagir com os níveis através do movimento da moldura e a seleção de cada carta.

No menu final, ao pressionar a tecla P (Play again) o jogador começa um novo jogo na dificuldade previamente selecionada e ao pressionar a tecla M (Main Menu) o jogador pode voltar ao menu principal para trocar a dificuldade ou sair do jogo.

Através da tecla B (Back) o jogador consegue voltar ao menu anterior e através da letra Q (Quit) consegue sair do jogo por completo.

3.4. Mouse

O mouse é igualmente importante ao teclado visto que todas as funcionalidades do teclado podem ser replicadas através do movimento do rato e dos três botões inerentes ao rato.

Usando o rato, no menu inicial é possível o jogador selecionar os níveis ou sair do jogo carregando com o botão esquerdo na escolha desejada; durante o jogo os botões do meio e direito do rato servem a funcionalidade de retroceder ao menu anterior, enquanto o botão esquerdo serve para selecionar a carta desejada para virar; no menu final o jogador pode clicar com o rato esquerdo em “Play Again” se quiser jogar um jogo da mesma dificuldade outra vez ou clicar em “Main Menu” se desejar voltar ao menu principal.

O jogador pode assim escolher o modo de como jogar e interagir com os menus com o teclado, mouse ou usando os dois ao mesmo tempo.

3.5. Vídeo

Todo o jogo passa por este dispositivo visto que é este usado para desenhar todas as imagens e animações desejadas durante a execução do jogo (todas geradas através de xpm's e passadas para sprites). O modo usado foi 0x14C, com 1052x864 de resolução e 16 777 216 cores em direct color mode (4 294 967 296 se contarmos com todas as opções de transparência).

Double buffering com recurso a cópia de memória foi usado de modo a tornar o jogo mais fluido; este foi implementado com recurso a dois buffers atualizados a cada interrupção do timer.

As sprites animadas foram usadas na parte de trás das cartas do jogo. Quando o rato está em cima das cartas ainda não viradas, a animação é ativada, dando a ideia que o dinossauro está a correr.

Foram usadas fontes para imprimir o tempo que o jogador está a demorar a resolver o jogo atual.

4. Estrutura do Código

Módulo	Percentagem (%)
Timer	5
Keyboard	15
Mouse	15
Graphics	15
View	25
Model	15
Sprites	5
Utils	5

4.1. Timer

Contém as funções previamente definidas em aula no lab2 que são necessárias para a implementação de interrupções. Inclui o ficheiro i8254.h que é indispensável devido a definição das macros usadas no módulo.

4.2 Keyboard

O módulo keyboard faz uso das funções implementadas no lab3 desenvolvidas no contexto de aula. Inclui o ficheiro i8042.h que contém as macros para o keyboard e também para o mouse.

4.3 Mouse

O Mouse foi implementado usando uma versão um pouco mais simplificada das funções do rato lecionadas no âmbito do Lab4. Como dito anteriormente, o módulo faz uso do ficheiro i8042.h com as macros necessárias e partilhando assim com o teclado.

4.4 Graphics

O módulo de gráficos foi um pouco simplificado e adaptado às necessidades do projeto, usando como base as funções exploradas no lab5.

4.5 Model

Dentro do módulo Model (parte de um sistema MVC) estão presentes as máquinas de estados que permitem ao jogo de processar as diferentes seleções dependendo de onde se encontra o jogador e processa os inputs deste. Neste módulo também é atualizado o estado do timer, mouse e são criados e eliminados os sprites.

4.6 View

Dentro do módulo View (parte de um sistema MVC) é onde a lógica do jogo foi implementada e este é desenhado na sua totalidade. Dentro deste módulo são desenhados os menus, os backgrounds, as cartas e as animações. Os diferentes níveis são desenhados e a função game é implementada que engloba a todos os níveis todas as funções criadas.

4.7 Sprites

Contém código relativo à criação de sprites através de xpm's e a destruição destas. No total o jogo conta com 24 sprites durante toda a sua execução.

4.8 Utils

Utils é o módulo usado desde o início da cadeia onde foram criadas as funções de retorno do bit mais significativo, do bit menos significativo e a função crucial `util_sys_inb`.

5. Detalhes de implementação

5.1. Máquina de Estados

```
typedef enum { MENU_START, MENU_INGAME, MENU_END } Menu;  
typedef enum { RUNNING, EXIT } System;  
typedef enum { EASY, MEDIUM, HARD } Difficulty;  
typedef enum { NEW, KEEP } Frame;  
typedef enum { DO, DONE } Setup;  
typedef enum { FIRST_CHOICE, SECOND_CHOICE, PAUSE, AFTER_PAUSE } GamePhase;
```

Neste projeto recorreremos ao uso de uma máquina de estados de modo a ser facilmente acedido o estado em que se encontra o jogo. Assim sendo a execução das funções varia dependendo do modo em que se encontra como por exemplo o enum Menu que tem MENU_START, MENU_INGAME e MENU_END que representa os três modos principais do jogo sendo respectivamente o menu inicial onde é selecionada a dificuldade do jogo, o menu de jogo onde é possível a execução deste mesmo e o menu final após completar o jogo que congratula o jogador pela vitória e permite que este jogue de novo ou volte ao menu inicial. Temos um enum que permite a fácil distinção de modo de dificuldade de modo a quando o jogo for chamado o nível irá ser adequadamente selecionado. Temos por fim uma máquina de estados que determina se a escolha do utilizador foi a primeira ou a segunda de modo a que seja mais fácil desenvolver as interações diferentes embora similares que acontecem quando é o primeiro movimento ou o segundo.

5.2. Modelo MVC

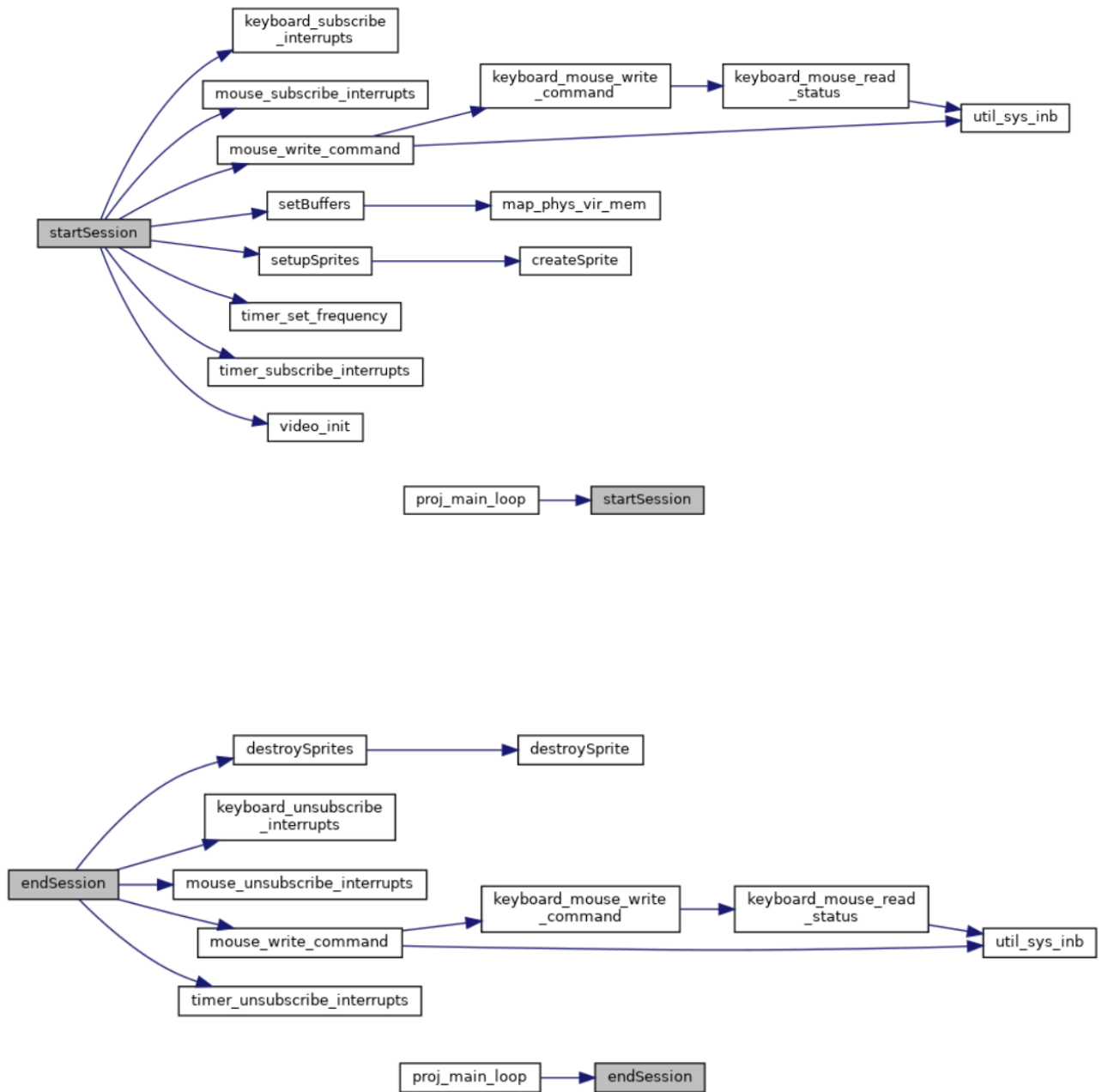
Decidimos implementar o código seguindo a arquitetura MVC (Model-View-Controller) de modo a haver uma criação de código mais intuitiva separando o projeto em 3 componentes com propósitos distintos. Assim sendo usamos o Model para aceder, manipular e armazenar dados. No modo View é desenhado o jogo com o objetivo em mente de ser o que o jogador verá durante o jogo e também implementar a lógica do jogo. Através deste o jogador é apresentado de várias opções de input para serem processadas. Finalmente o Controller processa as interações do utilizador recolhidas pelo com o jogo através do View de modo a atualizar o Model para mudar para o estado necessário para a execução do projeto.

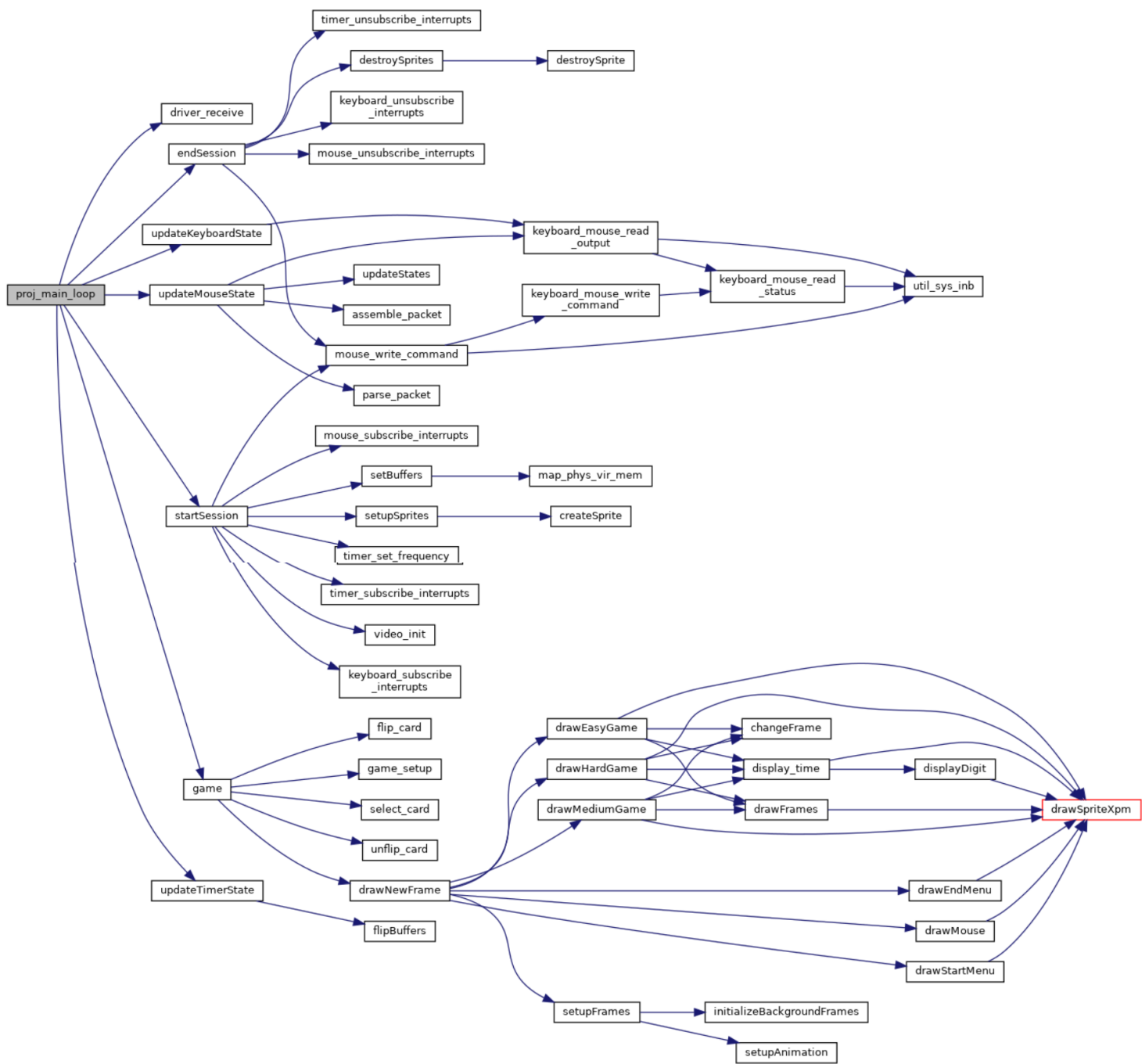
5.3. Animações

De modo a oferecer uma experiência interativa ao jogador decidimos que seria uma maneira de usar as interrupções do sistema para causar uma mudança entre as várias frames da carta virada para baixo de modo a criar uma animação do dinossauro inspirado no jogo da Google T-Rex Runner. Assim sendo quando o jogador movimenta o rato para cima de uma carta voltada para baixo o dinossauro é animado de modo a parecer que este está a movimentar-se.



6. Gráficos de Chamadas de Funções





7. Conclusões

No decorrer da unidade curricular Laboratório de Computadores deparamo-nos com várias dificuldades e percalços especialmente numa fase inicial, no entanto, esta cadeira provou ser bastante enriquecedora, embora trabalhosa. O funcionamento por labs foi uma maneira de trabalhar autonomamente e ao completar o projeto emanou no grupo um sentimento de satisfação por conseguir realizar um projeto que todos nos orgulhamos.

Como sugestão sentimos como grupo que podia ser dada uma maior quantidade de informação relativa aos periféricos de modo a que fosse mais fácil a integração de cada aluno a esta cadeira.

Finalmente, tivemos como já dito previamente vários obstáculos desde a execução de periféricos mais complexos como a gráfica e o rato e também na detecção de alguns erros e bugs que não são pontuais no Minix, para além de ser muito difícil fazer debug no Minix.