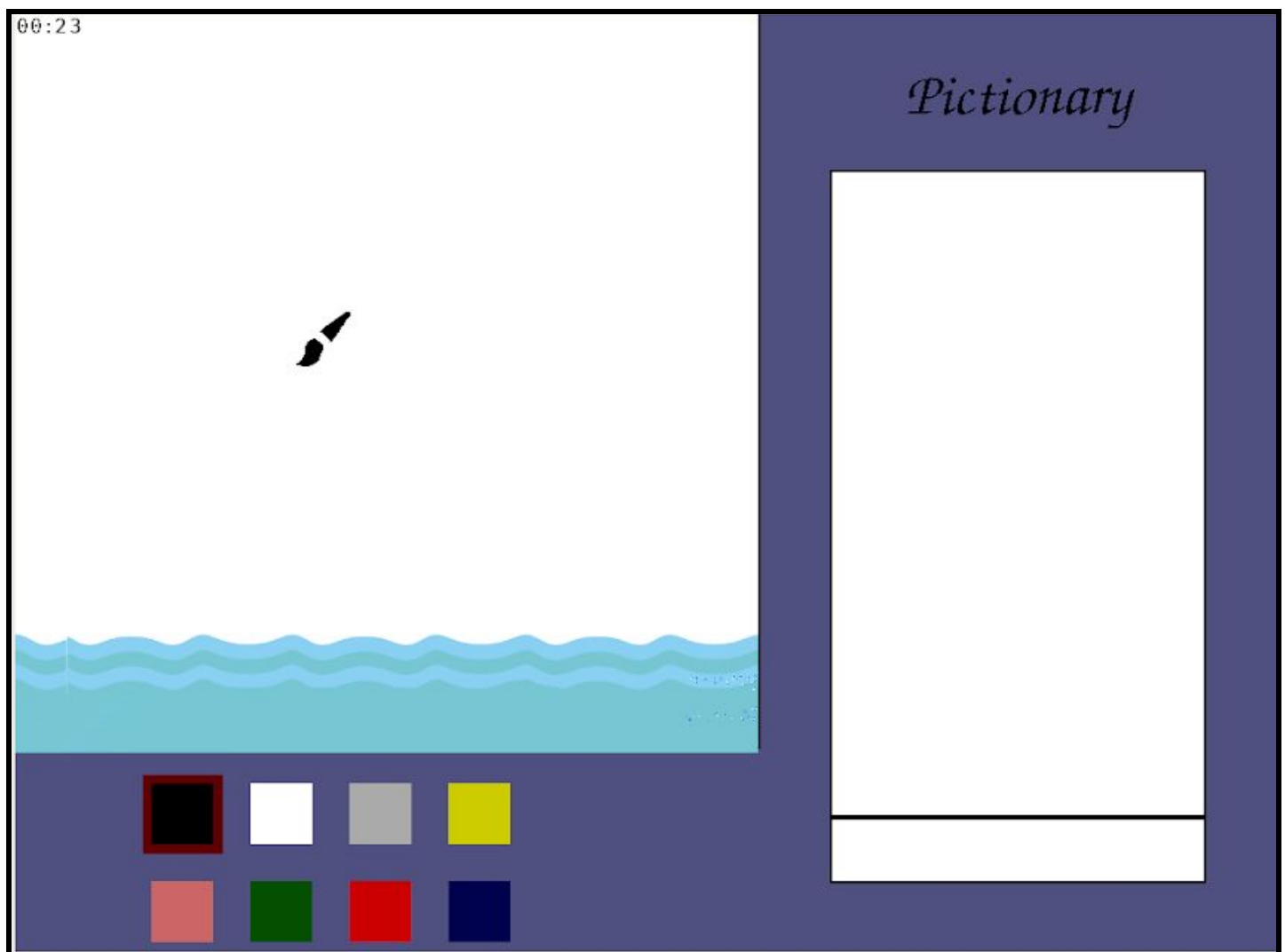


Pictionary

Minix



Trabalho realizado por:

Andreia Barreto Gouveia

Gonalo Fernandes Pereira

up201706430@fe.up.pt

up201705971@fe.up.pt

Índice:

1. Instruções de Utilização do Programa	-----	3
2. Estado do Projecto	-----	14
3. Estrutura/Organização do Código	-----	17
4. Detalhes de implementação	-----	23
5. Conclusões	-----	24

1.Instruções de Utilização do Programa

1.1.Objetivo do jogo

O jogo que desenvolvemos foi feito com o propósito de ser exclusivamente multiplayer. Este consiste no adivinhar uma palavra (tarefa desempenhada pelo jogador2) através de um desenho (realizado pelo jogador1). Quanto maior for o número de palavras certas, mais pontos serão ganhos

1.1. Menus existentes no jogo

Início

No início do programa o utilizador é apresentado a um ecrã de Boas Vindas de apresentação que inclui o nome do jogo. Para iniciar o jogo, o utilizador terá de carregar na tecla “Enter”.

Pictionary Minix

Iniciar

Menu de escolha de dificuldade

De seguida, é apresentado um menu que permite ao utilizador, seleccionar o nível de dificuldade do jogo, sendo esta opção permanente para o resto do jogo, sendo que para a modificar, o jogador terá de desligar e voltar a ligar o jogo.

As opções disponíveis são:

- “EASY”
- “MEDIUM”
- “HARD”

Estas poderão ser seleccionadas utilizando o rato ou as setas do teclado e, quando o botão esquerdo do rato é pressionado ou quando é pressionada a tecla “Enter”, a opção pretendida é inserida.

Os níveis de dificuldades possuem palavras diferentes, seleccionadas de modo a conformar com a dificuldade apresentada. A pontuação também difere de nível para nível, diminuindo com uma maior complexidade das palavras. Ou seja, esta decrementa consoante a maior dificuldade do nível escolhido, sendo que o nível “Hard” não só possui uma menor pontuação em relação aos outros níveis por palavra acertada, como também uma variedade de palavras muito mais complexa.

SELECT YOUR DIFFICULTY LEVEL

EASY

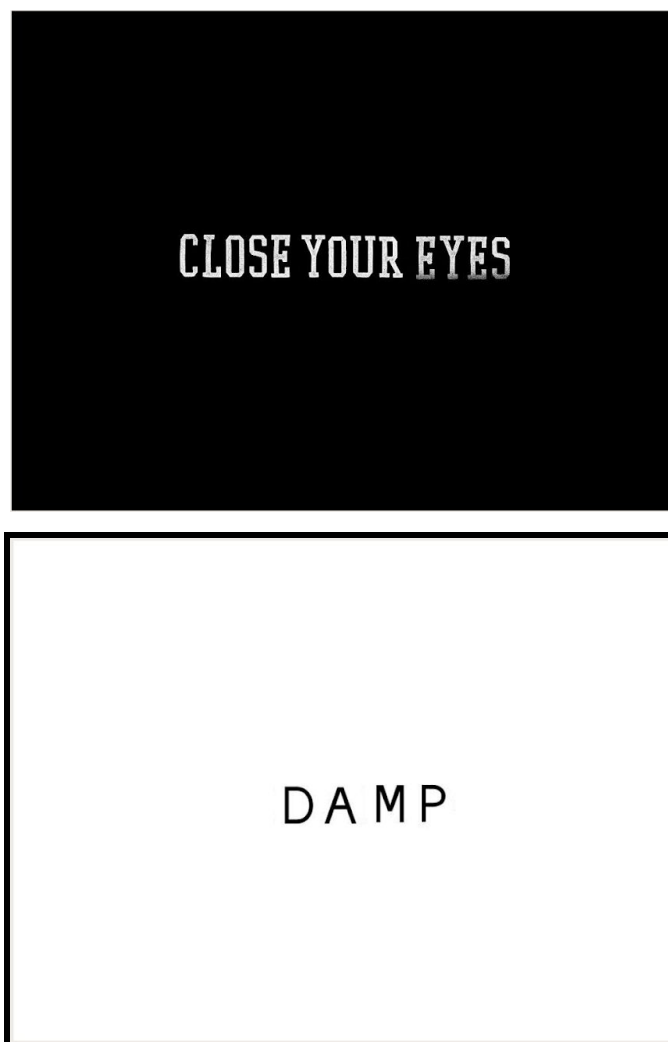
MEDIUM

HARD

Revelação da Palavra Escolhida

A escolha da palavra é feita de modo aleatório. Num primeiro instante é pedido ao jogador2 para “fechar os olhos” (“CLOSE YOUR EYES”), pois a palavra irá ser apresentada.

O primeiro jogador irá poder ver a palavra e terá 3 segundos para pensar no que desenhar de melhor modo a representar a palavra que lhe foi apresentada. Isto é feito numa contagem decrescente, como se pode observar nas figuras em baixo.



3

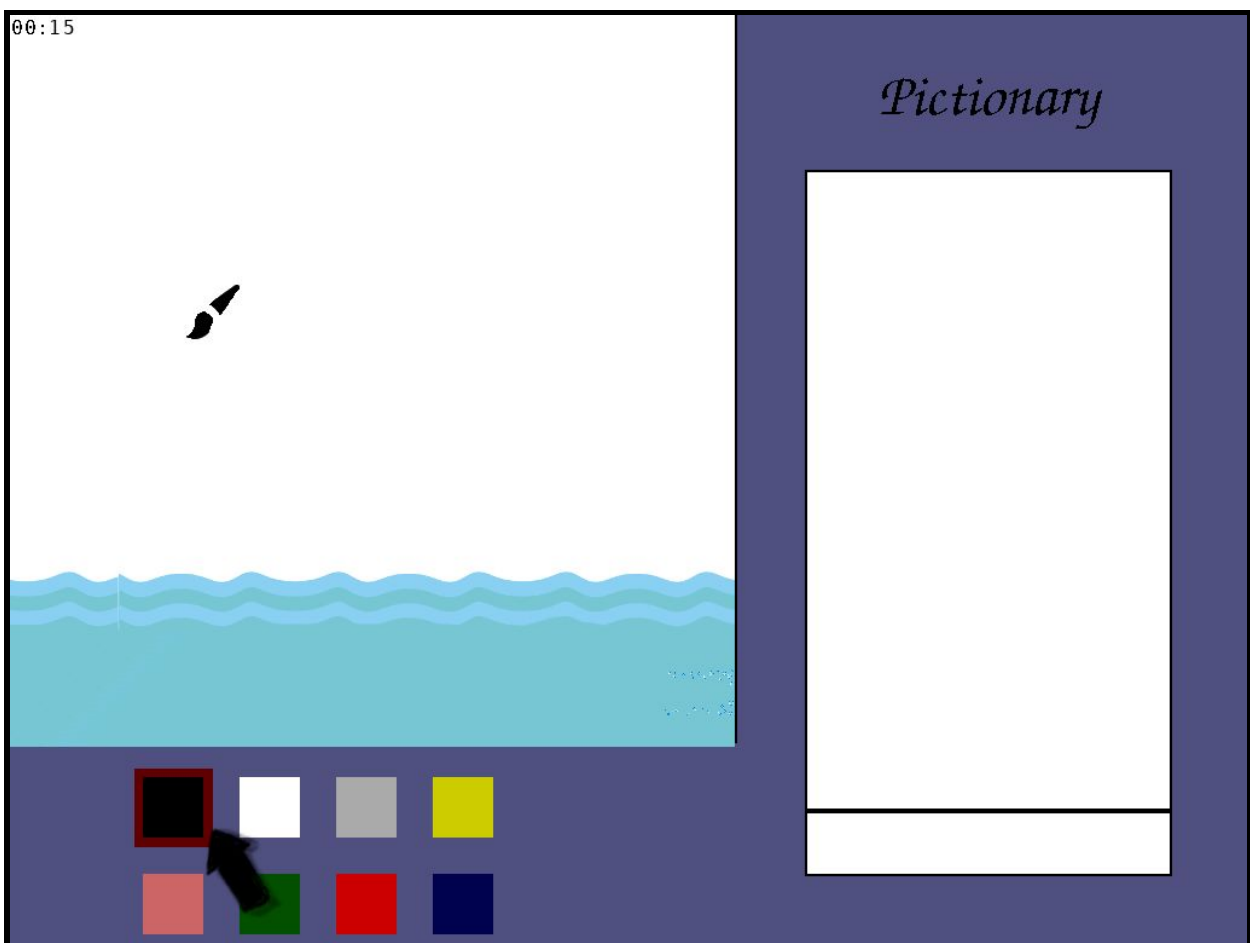
2

1

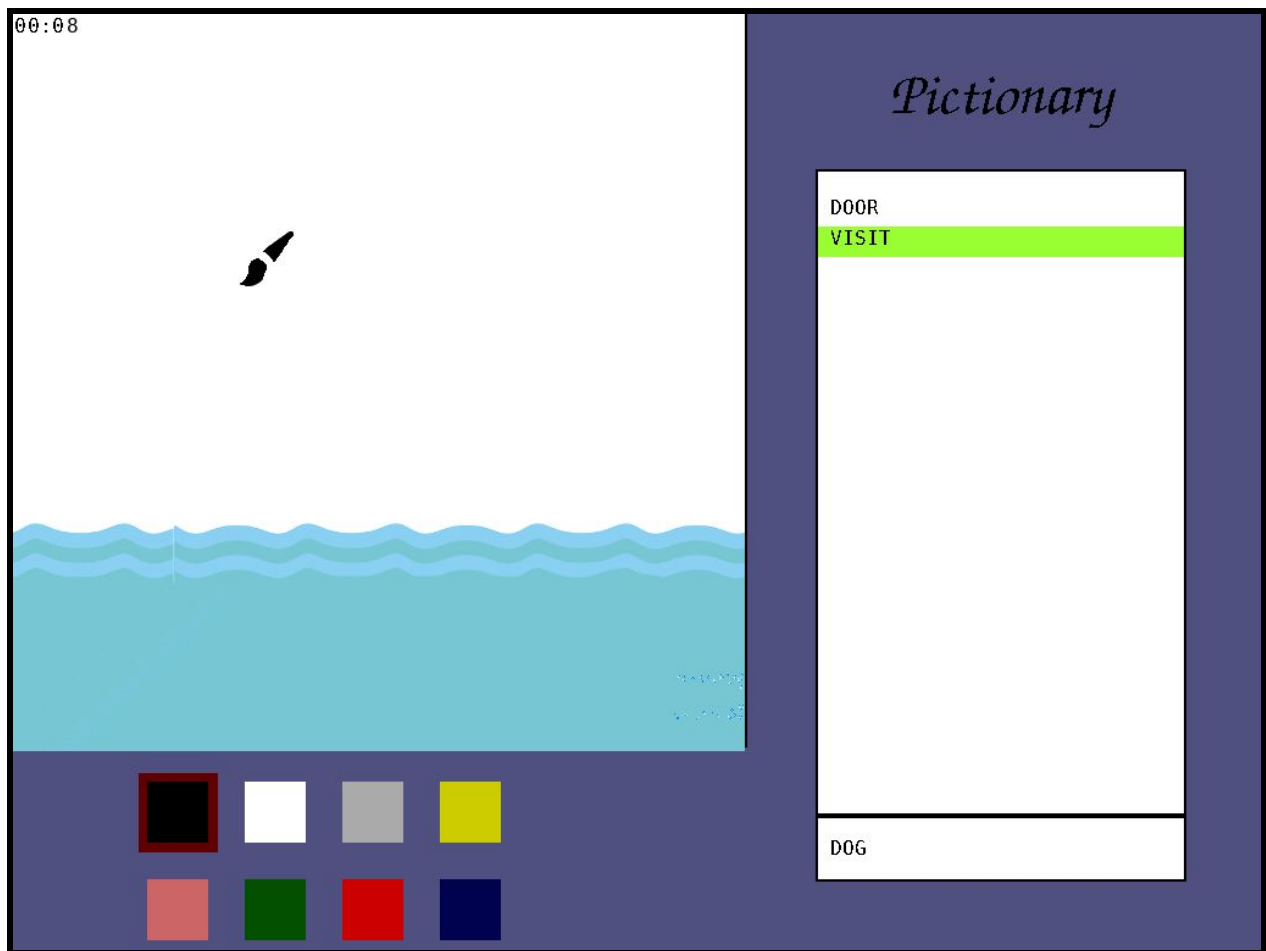
Menu de “Desenho”/Jogo

Neste menu, o jogador1 tem um painel onde pode desenhar pressionando o botão esquerdo do rato.

Imediatamente em baixo deste painel de desenho (que está identificado na imagem em baixo com uma seta) o jogador tem a opção de escolher a cor da linha que pretende, podendo a alterar a qualquer momento do jogo, utilizando as setas do teclado. O jogador sabe qual a cor selecionada pela indicação visual de bordas vermelhas em volta da cor em questão.



Ao lado desta seleção de cores, a tabela branca, será onde o jogador2 irá escrever e submeter as suas tentativas. Para submeter uma tentativa, o jogador terá de escrever a sua resposta e depois pressionar a tecla “Enter” e se estiver correta, a resposta irá aparecer dentro de um retângulo verde, indicando ao jogador que a resposta está correta e que a próxima palavra estará a surgir.



No canto superior esquerdo ambos os jogadores poderão visualizar um temporizador. À medida que o temporizador decresce, a “onda” ,que está visível no painel onde o jogador1 desenha, vai incrementando o seu tamanho. Esta onda implica que o jogador1 não pode desenhara na parte do ecrã onde a mesma está representada, nem ver o que foi anteriormente desenhado, ou seja, com o passar do tempo, vai reduzindo a área disponível para desenhara.

Menu de Pausa

O jogador ao pressionar a tecla “Esc” vai lhe ser apresentado um menu que lhe permite escolher se quer continuar no jogo ou sair do mesmo.

As opções disponíveis são:

- “NO”
- “YES”

Ao seleccionar a opção “NO”, ou seja, continuar o jogo, o jogo é restaurado ao estado em que este se encontrava. No entanto, se seleccionada a opção “YES”, ou seja, se o utilizador pretender sair do jogo, a pontuação é restaurada a 0 (pois não terminou o mesmo) e imediatamente a seguir é apresentado o menu de novo jogo.



Menu de Novo Jogo

Quando o jogador decide sair do jogo antes do mesmo terminar ou quando o temporizador chegar a 0, depara-se com o menu de novo jogo, onde pode decidir entre começar um novo jogo ou sair permanentemente do mesmo.

Novamente, para seleccionar a opção, poder-se-á utilizar tanto o rato ao pressionar o botão esquerdo, tal como o teclado, pressionando a tecla “Enter”

As opções disponíveis são:

- “NEW GAME”
- “EXIT”

Se o jogador seleccionar a opção “NEW GAME” irá ser direccionado para um novo jogo, tendo a oportunidade de voltar a tentar superar o highscore.

Por outro lado se seleccionar “EXIT”, o jogo encerrará.



Menu de Notificação Novo HighScore

Logo após o jogo ter terminado com o temporizador com valor zero e se a pontuação adquirida ao longo do jogo, for maior que o highscore irá aparecer uma breve animação que indica ao utilizador que atingiu o novo highscore, e de seguida o mesmo poderá visualizar a sua pontuação no ecrã.



Menu de novo Score

Aparece no ecrã a nova pontuação adquirida pelo jogador ao longo do jogo.

SCORE:

4 4

2.Estado do jogo

A tabela seguinte apresenta os periféricos utilizados no jogo, uma breve descrição prática da sua implementação e se usaram interrupções.

Periférico	Utilização	Interupções
Timer	-Atualização de animações; -Atualização do temporizador; -Sincronização da apresentação do temporizador,brush e desenho;	Sim
Keyboard	-Seleção de opções; -Escrita de texto no ecrã;	Sim
Mouse	-Seleção de opções; -Desenho;	Sim
Graphics	-Visualização do jogo;	Não
RTC	-Consulta da Data para guardar a pontuação num ficheiro txt;	Não
Serial Port	---	---

TIMER

Este periférico é utilizado para:

- Atualizar a imagem apresentada na animação do highscore
(função : `show_score()/draw_bongo()`, localizado no `proj.c`)
- Atualizar o counter do temporizador
(função : `game()` , localizado no `proj.c`)
- Sincronização da apresentação do temporizador,brush e desenho
(função : `game()` , localizado no `proj.c`)

Este periférico foi implementado no `proj.c`, `timer.c` e `timer.h`.

KEYBOARD

Este periférico é utilizado para:

- Selecionar opções em todos os menus apresentados
- Escrever texto e inserir respostas
(função : `game()` , localizado no `proj.c`)
- Permite fazer pausa num jogo
(função : `game()` e `continue_game()` , localizado no `proj.c`)

Este periférico foi implementado no `proj.c`, `keyboard.c` e `keyboard.h`.

GRAPHICS CARD

O modo utilizado é o `0x118` ,cuja resolução é de `1024x768`, com o modo `RGB 8:8:8` (`16777216` cores)

Este periférico é utilizado para:

-Visualizar todas as componentes do jogo, ou seja, os menus, a linha a ser desenhada, às palavras a serem escritas e submetidas, à escolha de opções e animações.

É também relevante mencionar que é utilizado o “double buffering”, onde são utilizados os seguinte buffers:

- `Video_mem;`
- `Drawing_buff;`
- `Background_buff`

Graças a este método foi possível fazer uma abordagem por camadas no que toca a apresentar o fundo, o brush e os menus.

Uma das aplicações destes buffers foi por exemplo no que toca a guardar o conteúdo do ecrã. O Background_buff armazena o layout do jogo em si mais as linhas que vão sendo desenhadas e as palavras que vão sendo usadas como tentativas; o Drawing_buff é uma cópia do Background_buff sobreposto pelo brush e pelo timer. Depois de ter a imagem montada e pronta, esta é finalmente enviada para o video_mem, que será apresentada no ecrã. A razão para usarmos o Drawing_buff como um intermediário foi a de evitar que apareçam “flashes” e imagens corrompidas no ecrã, uma vez que pode ocorrer por vezes de as imagens não serem processadas totalmente e simultaneamente.

Para além disso, este método trouxe também melhoria no que toca à fluidez gráfica do jogo

Para escrever no ecrã e demonstrar as inúmeras imagens, utilizamos todas as letras do alfabeto, números e o ‘:’. Os menus, tal como o ecrã de apresentação do jogo estão em imagens com texto.

Foi também implementado um sistema de colisões no que toca à navegação nos menus, sendo que quando o rato passa sobre uma opção esta é realçada com um traço branco em baixo da opção, e no caso de se fazer uso do botão esquerdo do rato, esta é selecionada.

Foram também usadas colisões nas bordas da área de desenho do brush.

MOUSE

Este periférico é usado para:

- Seleção de opções
(todos os menus, localizados no proj.c)
- Desenhar quando pressionado o botão esquerdo
(função : game(), localizado no proj.c)

Este periférico foi implementado no proj.c, mouse.c e Mouse.h.

RTC

Este periférico é usado para:

- Consulta da data para guardar a pontuação num ficheiro txt;

(função : main(), localizado no proj.c)

Este periférico foi implementado no proj.c, rtc.c e rtc.h.

3.Estrutura/Organização do Código

Módulos:

Módulo proj.c

-Aqui é onde se encontra todo o jogo. É neste ficheiro que o jogo ganha forma, sendo um dos ficheiros mais importante do projecto.

-Os vários menus são evocados no main deste projecto (game(),continue_game(),menu(),initial_mode(),draw_bongo(),show_score(),new_game()) sendo alguns destes ligados por um ciclo while que permite ao utilizador continuar a jogar, nomeadamente o game(), new_game() .

-Associado ao game está o continue_game(), que vai dar origem ao ecrã de pausa. Isto é, quando pressionada a tecla “Esc”, direcciona o utilizador para o menu de pausa. Sendo que se este quiser permanecer no jogo, este é lhe restaurado.

-A função show_score() é apresentado quando um jogo termina com o temporizador a 0, sendo que, é apresentado no ecrã o valor da pontuação acumulada ao longo do jogo.

-Este módulo foi igualmente desenvolvido por ambos os membros.
Peso do módulo no projecto: 35%

Módulo mouse.c

-Aqui encontram-se as funções que se encarregam de ler as interrupções do rato e formar os diversos packets que depois são interpretados no proj.c. Todas estas funções foram reutilizadas do laboratório que desenvolvemos este semestre.

Este módulo foi igualmente desenvolvido por ambos os membros.
Peso do módulo no projecto: 6.25%

Módulo score.c

-Aqui é onde a pontuação é lida, a partir de um ficheiro denominado de “score.txt”(readHighScore()). Depois do jogo terminar na sua totalidade, a informação do highscore é guardada no ficheiro “score.txt” a partir da função store_information(). Note-se que quando a informação é guardada, a data e hora que quando mesma foi gerada também é apresentada. Por último, a função find_number_of_digits() é uma função que é útil na escrita da pontuação do ecrã, pois ajuda a calcular uma posição inicial mais correta para tal.

-Este módulo foi desenvolvido pela Andreia Gouveia.

Peso do módulo no projecto: 3%

Módulo keyboard.c

-As interrupções do keyboard vão ser tratadas neste módulo, sendo que aqui é onde os scan codes do keyboard vão ser interpretados e posteriormente enviados para o proj.c. É aqui também onde foram criadas funções auxiliares para converter o scan code para código ASCII (ScanToAsc() que foi desenvolvida pelo Gonçalo Pereira) , o resto foi desenvolvido igualmente por ambos os membros.

Peso do módulo no projecto: 6%

Módulo RTC.c

-O RTC (real time clock), tem como propósito o fornecer a data (e horas) de quando um jogo acaba, sendo essa informação posteriormente guardada no “score.txt”.

-Como a informação vem em hexadecimal, foi necessário a criação de uma função que permitisse a conversão para um número decimal e tal foi feito na função convert() que se encontra definida no ficheiro rtc.c.

-Este módulo foi desenvolvido pela Andreia Gouveia, excetuando a função printDate() que foi implementada pelo Gonçalo Pereira.

Peso do módulo no projecto: 3%

Módulo timer.c

-Este ficheiro não foi alterado para este projeto. Todas as funções já foram desenvolvidas no laboratório correspondente, este semestre. No entanto foram algumas funções removidas pois não fazia sentido estarem no nosso trabalho visto que não possuíam utilidade.

-Este módulo foi igualmente desenvolvido por ambos os membros.

Peso do módulo no projecto: 6%

Módulo video.c

-Este ficheiro é o segundo ficheiro mais importante do projecto todo. É aqui onde as linhas são desenhadas (`draw_line()`), os menus e cursor exibidos, as opções selecionadas e é onde é notório o uso do triple buffering, aspeto essencial para a fluidez do trabalho.

-Este módulo foi igualmente desenvolvido por ambos os membros.

Peso do módulo no projecto: 30%

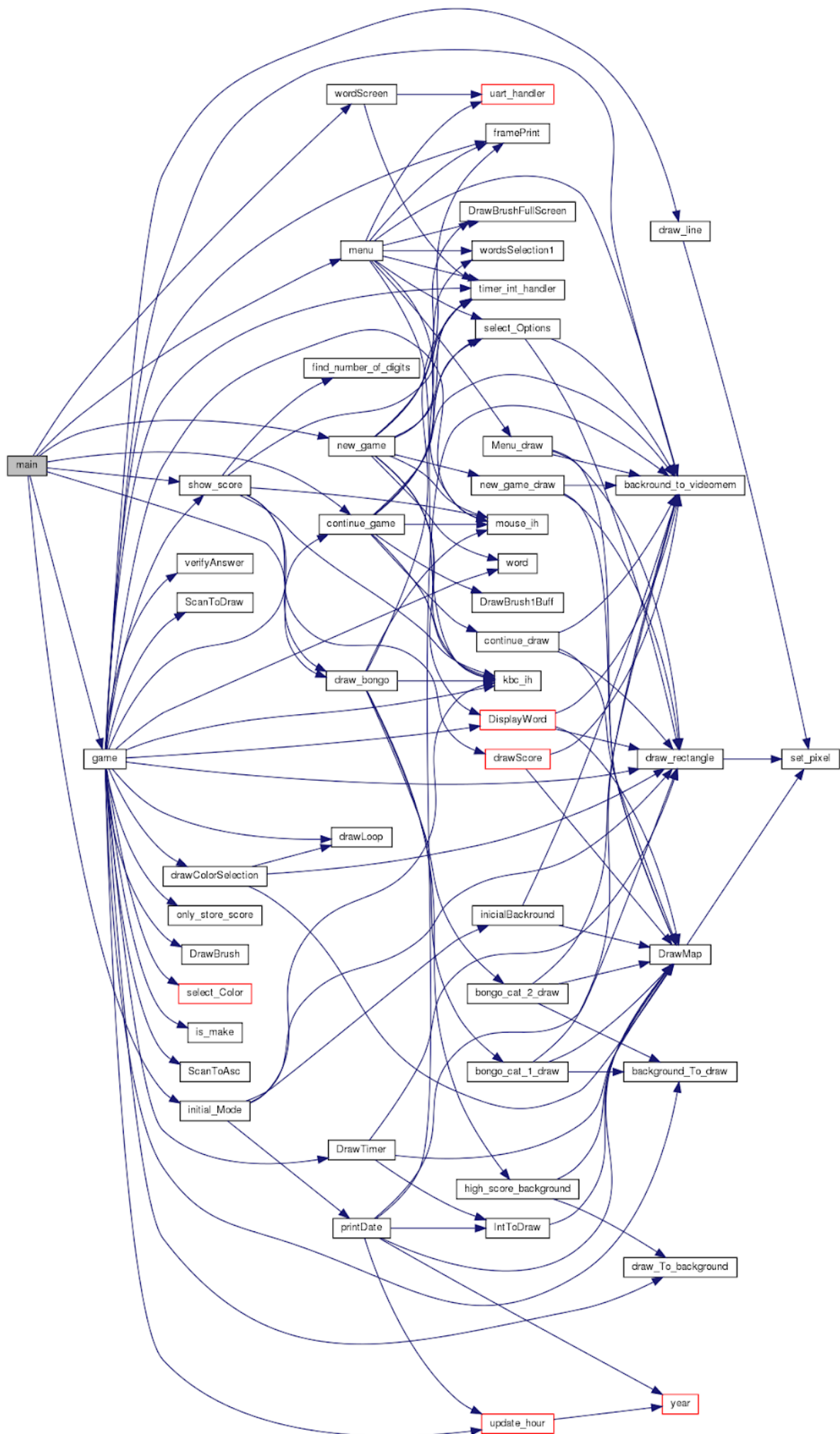
Módulo words.c

-Aqui é onde, dependendo da dificuldade selecionada, que o conjunto de palavras é escolhido (`wordsSelection1()`). É escolhida uma palavra (função `word()`) de forma aleatória, através de funções como o `rand()`, e é essa palavra que o jogador 2 terá de tentar adivinhar. A verificação de que se a palavra que o jogador 2 está correta ou não é feita na função `verifyAnswer()` (desenvolvida pelo Gonçalo Pereira).

-O restante deste módulo foi desenvolvido pela Andreia Gouveia

Peso do módulo no projecto: 6%

Function Graph:



4.Detalhes de Implementação

Neste trabalho possuímos colisões, isto em todos os menus, na seleção de opções através do mouse, em que , quando o cursor se encontra dentro de um determinado intervalo x e y, a barra (branca que se encontra por de baixo de cada palavra. É o indicador visual de que uma palavra está selecionada) mudará consoante a opção que o rato seleccionar(mas , ao desenhar, o cursor não consegue o fazer para além da zona delimitada pela onda, nem consegue sair da caixa onde esta atividade é realizada).

Usamos programação orientada a objetos no RTC ,a struct info, que cria um objecto que contém o dia, o mês, o ano , a hora, o minuto e o segundo que uma nova pontuação foi criada.

5. Conclusões

Achamos a disciplina LCOM, algo que requer muito trabalho e muitas horas dedicadas, mas ,contudo, sentimos que nos forneceu bases muito importantes para programação de baixo nível, mostrando também novas perspectivas e aprofundando o nosso conhecimento sobre tais periféricos.

A nível teórico, achamos que a disciplina conseguiu transmitir bem a informação (principalmente complementando com as aulas teóricas) , no entanto a nível prático achamos que houve uma discrepância e alguma falta de disponibilização de informação. Mas no entanto, a disponibilização de monitores para nos ajudar nos trabalhos foi algo que nos ajudou a superar tal problema, tal como as dúvidas no moodle.

No entanto, achamos que o serial port precisava de ser praticado numa aula prática, pois é um dispositivo de difícil implementação, o qual tivemos algumas dificuldades.