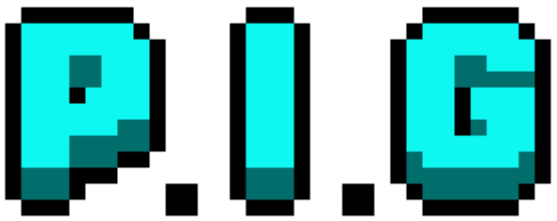


Projeto Final
Laboratório de Computadores
Turma 9 - Grupo 4

Realizado por :

- André Dantas Rodrigues: up202108721@fe.up.pt
- Afonso Campelo Poças: up202008323@fe.up.pt
- Miguel Bravo Almeida Silva Figueiredo: up201706105@fe.up.pt

ÍNDICE

Introdução

O nosso projeto P.I.G (Pixel Image Guesser) é uma adaptação do jogo picross. O objetivo deste é adivinhar o correto posicionamento dos pixels coloridos utilizando as pistas fornecidas.

Instruções de utilização

Menu inicial :

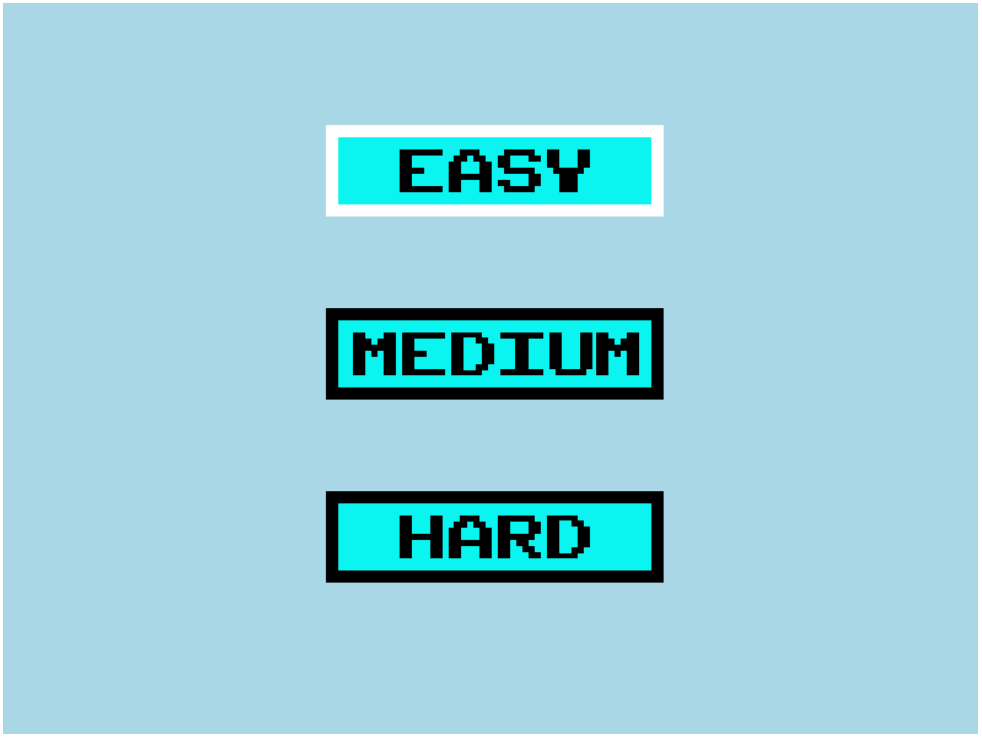
Neste menu inicial podemos selecionar entre jogar , ver as regras do jogo e sair do jogo. Para selecionar uma opção o

utilizador pode usar o keyboard usando W e S para mudar a opção pretendida. Para a selecionar basta premir SPACE.

Instruções :

Nesta janela mostramos as regras do jogo. Para sair desta janela o utilizador precisa de clicar na tecla Esc.

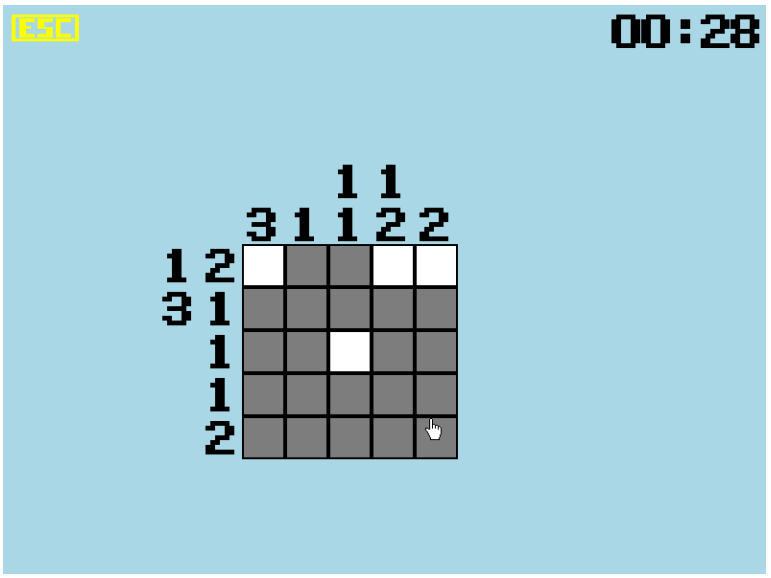
Selecionar dificuldade :



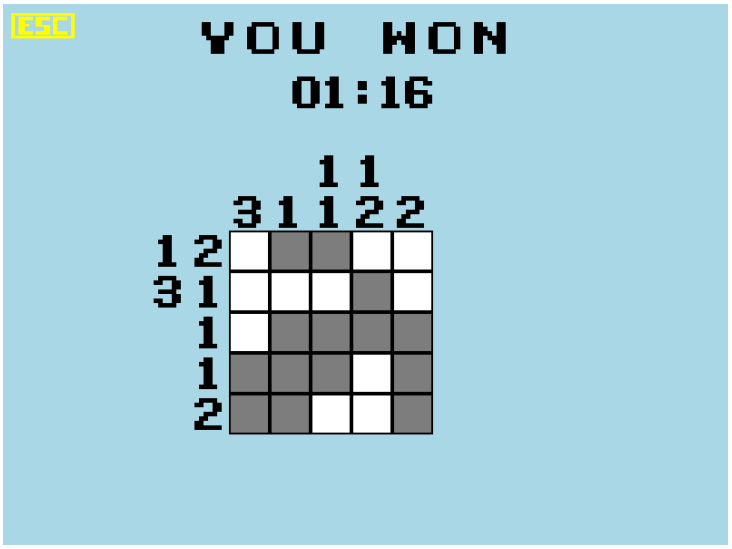
Tal como no menu anterior o utilizador pode navegar entre as opções disponíveis usando as teclas W, S, ESC e SPACE, tendo a opção de escolher três dificuldades distintas:

- Easy: tabuleiro 3 por 3;
- Medium: tabuleiro 5 por 5;
- Hard: tabuleiro 7 por 7;

Jogo :



Dentro do jogo, o utilizador pode usar o rato para colorir os pixels do tabuleiro. Quando o utilizador coloriu todos os pixels corretos o jogo termina, sendo mostrado o tempo demorado na resolução.



Estado do projeto

TABELA DE DISPOSITIVOS:

Dispositivos	Funcionalidades	Interrupções
TIMER	Frame-rate, tempo de jogo	SIM
KEYBOARD	Navegação de menus	SIM
MOUSE	Selecionar/colorir pixel	SIM
VIDEO GRAPHICS	Mostrar toda a interface com o utilizador: menus,board, hints e tempo do jogo	NÃO

TABELA DE FUNCIONALIDADES:

FUNCIONALIDADE	DISPOSITIVOS	ESTADO
Navegação de menus	KEYBOARD	COMPLETO
Escolha de diferentes dificuldades	VIDEO GRAPHICS, KEYBOARD	COMPLETO
Display da board	VIDEO GRAPHICS	COMPLETO
Display das hints	VIDEO GRAPHICS	COMPLETO
Display do tempo do jogo	VIDEO GRAPHICS, TIMER	COMPLETO
(Des)Colorir pixels	VIDEO GRAPHICS, MOUSE	COMPLETO
Display das instruções	VIDEO GRAPHICS	COMPLETO
Janela de Fim do jogo	VIDEO GRAPHICS	COMPLETO

DISPOSITIVOS

TIMER

O timer usado neste projeto foi o timer 0, tendo servido para mostrar o tempo demorado a resolver o puzzle e para ajustar a frame rate. Escolhemos 30 frames por segundo de modo a não sobrecarregar o sistema e evitar crashes, mantendo na mesma uma experiência responsiva dos diversos elementos desenhados.

A implementação do seu handler encontra-se no ficheiro game.c (no caso do tempo de jogo), sendo usado no ficheiro proj.c para gerar as interrupções necessárias para desenhar o ecrã.

KEYBOARD

O teclado é usado neste projeto para interagir com os menus e mover entre estados de jogo, sendo usado as teclas W, S e SPACE para navegar os menus e a tecla ESC para voltar ao estado anterior bem como fechar o programa. Os seus handlers encontram-se nos ficheiros game.c, menu.c e level.c.

MOUSE

O nosso programa utiliza tanto os botões como a posição do rato para interagir com o tabuleiro de jogo. Usamos o botão esquerdo para “ligar” e “desligar” os pixels do tabuleiro, sendo necessária a posição do rato para detectar qual deles deve ser ligado.

A implementação do seu handler encontra-se no ficheiro game.c, nomeadamente as funções mouse_game_handler e draw_game_mouse.

VIDEO GRAPHICS

Usamos video graphics para dar display a todo o tipo de interfaces, tal como o menu e a board.
Usamos o modo direto 8:8:8 (0x115) com 800x600 de resolução (16 777 216 cores). Dispomos de double buffering e utilizamos draw_graphics_content() para passar o conteúdo do buffer para a VRAM. Damos display aos caracteres através de XPMs e recorremos a sprites para desenvolver os elementos das páginas.

Organização/estrutura do código

Dividimos o nosso código em model e controller, tentando emular a conhecida estrutura MVC (model, view, controller), apenas diferindo na junção do model com a view.

Na pasta controller encontra-se a implementação dos dispositivos I/O criados nos labs com algumas modificações (de acordo com as necessidades do projeto).

Já na pasta model temos a implementação das estruturas de dados usadas bem como os seus métodos, estando incluído ainda as funções referentes ao display tanto dos menus como do jogo em si, para além dos respectivos handlers.

De seguida encontram-se os módulos usados no projeto:

- **proj.c - 10%**

Onde se encontra a lógica do programa, contém 4 funções : start() onde se dá subscribe aos dispositivos I/O e se inicializa o modo de vídeo, loop() onde se gere a maior parte da lógica do programa e são chamados os handlers consoante o estado do programa, end() onde retornamos o minix ao seu estado inicial e finalmente temos o

proj_main_loop(), que assegura a correta ordem de execução das funções descritas.

- **game.c - 20%**

Apresenta handlers para os dispositivos I/O usados durante o jogo, para além de várias funções de dar display dos diferentes elementos do jogo. Apresenta também a função para verificar se o puzzle foi resolvido.

- **utils.c - 1%**

Ficheiro criado para os labs com algumas funções usadas na implementação dos dispositivos I/O.

- **graphics.c** - 5%

Contêm funções relacionadas com o vbe.

- **board.c - 10%**

Contêm as funções e estruturas relativas à criação e modificação do tabuleiro de jogo.

- **tile.c - 10%**

Contém funções e estruturas para a criação e manipulação de Tiles, que representam cada componente do tabuleiro.

- **kbc.c - 5%**

Contêm as funções usadas para a comunicação com o kbc (enviar/receber pacotes, comandos, etc).

- **mouse.c - 5%**

Contém funções relacionadas com o rato (receber pacotes, subscrever interrupções, etc).

- **keyboard.c - 5%**

Contém funções relacionadas com o teclado (receber pacotes, subscrever interrupções, etc).

- **timer.c - 5%**

Contém funções relacionadas com o timer (definir frequência, obter estado do timer, etc).

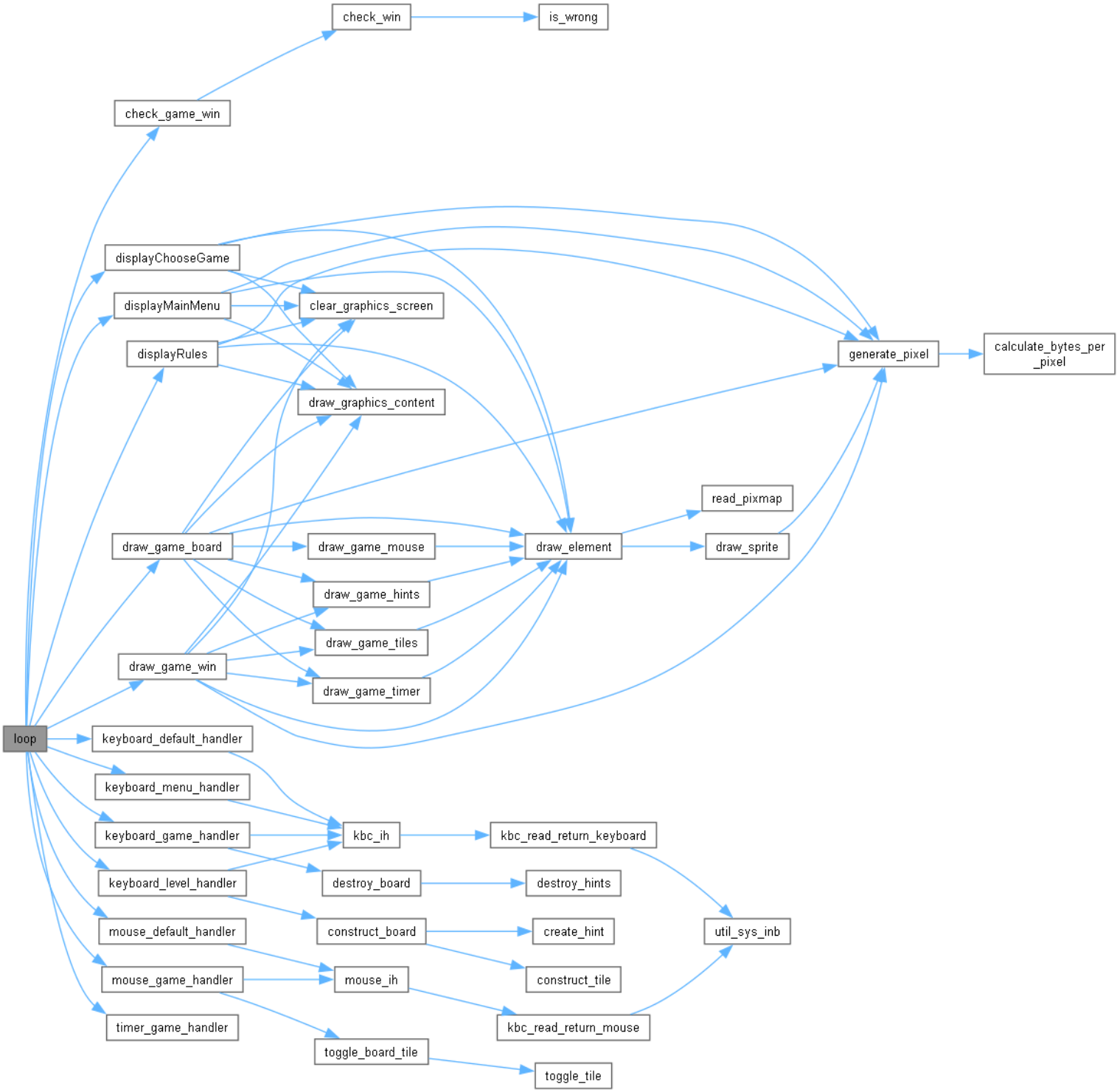
- **level.c - 10%**

Ficheiro onde se encontram os handlers dos diversos dispositivos usados no menu de escolher dificuldades, bem como funções para desenhar os elementos que o constituem.

- **menu.c -14%**

Ficheiro onde se encontram os handlers dos diversos dispositivos usados no menu principal, bem como funções para desenhar os elementos que o constituem.

Function Call Diagram



Conclusões

Este trabalho permitiu-nos interagir com dispositivos de baixo nível bem como proporcionou-nos uma oportunidade de praticar a programação destes, tendo sido implementadas grande parte das funcionalidades esperadas, embora tivéssemos que reduzir a escala de algumas destas devido a dificuldades de software. De qualquer forma, estamos contentes com o trabalho implementado.

Referências

https://www.piskelapp.com/p/create/sprite

https://www.online-utility.org/image/convert/to/XPM

https://www.shutterstock.com/pt/image-vector/vector-illustration-black-pixel-number-set-472896136?consentChanged=true