

**Faculdade de Engenharia da Universidade do Porto**



**Projeto Final**

**Laboratório de Computadores**

**Turma 2 - Grupo 4**

**Realizado por:**

- **Gonçalo Martins** ([up202108707@up.pt](mailto:up202108707@up.pt))
- **Diogo Viana** ([up202108803@up.pt](mailto:up202108803@up.pt))
- **David Cordeiro** ([up202108820@up.pt](mailto:up202108820@up.pt))

# Índice

Introdução.....	3
Instruções de Utilização.....	4
Menu inicial.....	4
Dentro do jogo.....	5
Fim de jogo .....	6
Estado do Projeto.....	7
Tabela de funcionalidades.....	7
Tabela de dispositivos.....	7
Dispositivos.....	8
Mouse.....	8
Teclado.....	8
Timer .....	8
Video Graphics .....	9
RTC .....	9
Organização do código/estrutura .....	10
Keyboard Module - 15% .....	10
Mouse Module - 10% .....	10
RTC - 5%.....	10
Timer - 10% .....	10
Video Card Module - 15% .....	10
Utils Module - 3%.....	10
Events Module - 12% .....	11
Game Auxilary Module – 15% .....	11
Sprites Module - 10%.....	11
Game Module - 5%.....	11
Function call Graph.....	12
Conclusões.....	13

# Introdução

O nosso Projeto *Cosmic Garden* consiste numa adaptação do jogo *Plants vs Zombies* para somente um jogador.

O jogo começa após o utilizador selecionar a opção *play* no menu principal. Após a inicialização, o jogador deve disparar e mover-se de forma a eliminar os *Zombies* que se dirigem para a personagem principal.

O jogo acaba quando é detetada uma colisão entre a *sprite* da personagem principal e a *sprite* de um dos *Zombies*.

# Instruções de Utilização

## Menu inicial

Ao correr o projeto, o menu inicial abaixo apresentado será a primeira página a aparecer no ecrã. Este contém 2 opções, 1. *PLAY* e *ESC. QUIT*, assim como a data atual. Ambas as opções podem ser selecionadas pelo teclado (1 e ESC, respetivamente) ou clicar com o botão esquerdo do rato sobre a área pretendida. Ao colocar o rato sobre a área da opção, a palavra da opção irá realizar uma pequena animação ao mudar de cor e posição.



## Dentro do jogo

Ao pressionar a opção 1. *PLAY*, o ecrã irá alterar para a figura abaixo apresentada.

O jogador irá conseguir movimentar a sua personagem, apresentada do lado esquerdo, com o teclado, ao pressionar as teclas *W*, *A*, *S* e *D*, movendo para cima, esquerda, baixo, direita, respetivamente. Para disparar, o jogador deve pressionar o botão esquerdo do rato. Os *Zombies* movimentam-se em “ondas”, o que significa que o jogador deve eliminar os primeiros 10 *Zombies* e, de seguida, aparecerão outros 10 *Zombies* que se irão movimentar ainda mais rápido que a “onda” anterior. Os *Zombies* são eliminados quando é detetada uma colisão entre o *sprite* do disparo e o *sprite* do respetivo *Zombie*.



## Fim de jogo

O jogo pode acabar de duas formas diferentes:

- É detetada uma colisão entre a personagem e um *Zombie*;
- Um *Zombie* chega ao lado esquerdo do ecrã.

Caso aconteça, o ecrã irá voltar ao menu inicial.

# Estado do Projeto

## Tabela de funcionalidades

Funcionalidades	Dispositivos	Estado de Implementação
Disparar	<i>Mouse e Video Graphics</i>	Completo
Movimento da personagem	Teclado e <i>Video Graphics</i>	Completo
Movimento dos <i>Zombies</i>	<i>Timer e Video Graphics</i>	Completo
Navegação entre menus	<i>Mouse, teclado e Video Graphics</i>	Completo
Demonstração da data atual	RTC e <i>Video Graphics</i>	Completo
Desenho das <i>sprites</i>	<i>Video Graphics</i>	Completo

## Tabela de dispositivos

Dispositivos	Funcionalidades	Interrupções
Mouse	Disparar, selecionar opções no menu, movimento no menu	Sim
Teclado	Selecionar opções no menu e mover personagem	Sim
Timer	<i>Frame Rate</i> , movimento de <i>Zombies</i>	Sim
<i>Video Graphics</i>	Mostrar interface com o jogador, desenho das <i>sprites</i> e movimento	Não
RTC	Demonstração da data atual	Sim

# Dispositivos

## Mouse

O mouse é usado para selecionar as opções do menu e para o disparo primário da personagem.

As diferentes posições do rato, transformadas depois em coordenadas cartesianas, permitem distinguir os botões do menu que o jogador seleciona.

No menu, o botão esquerdo do mouse é usado para selecionar a opção que o jogador pretende

A implementação das configurações, funcionalidades e interrupções do mouse estão no ficheiro `mouse.c`.

## Teclado

Tal como o mouse, o teclado também pode ser usado para selecionar as opções do menu. Para além disso, o teclado é utilizado para mover a personagem dentro do ciclo de jogo.

Em todas as nossas funções que utilizam o teclado fazemos o processamento do input através do *make code* gerado por cada tecla pressionada.

No menu inicial, podemos pressionar a tecla '1' para entrar no jogo e 'ESC' para fechar o programa.

Dentro do jogo, o *player* pode pressionar as teclas *W*, *A*, *S* e *D* para andar para cima, esquerda, baixo e direita, respetivamente, estando limitado a um espaço definido.

A implementação das configurações, funcionalidades e interrupções do teclado estão no ficheiro `keyboard.c`.

## Timer

O timer, mais concretamente o timer 0, é usado para limitar o tempo de cada jogo e para controlar a *frame rate* da placa gráfica, atualizando o ecrã 60 vezes por segundo.

A implementação das configurações, funcionalidades e interrupções do timer estão no ficheiro `timer.c`.



## Video Graphics

O *video graphics* é utilizado para dar display das várias páginas que criamos, bem como o jogo em si.

Todo o projeto está realizado no modo de vídeo, pelo que o display dos caracteres é efetuado através de XPMs. Usamos uma resolução de 1024x768 *pixels* (modo VBE 0x118), com cores de 3 bytes em modo direto e como tal permite  $2^{24}$  cores (16,8 milhões, aproximadamente) diferentes no nosso programa.

De forma a tornar o jogo mais fluido e dinâmico sob o ponto de vista do jogador implementámos as técnicas de *triple buffering* e *v-sync* mas, como o segundo apresentava um *delay*, não o utilizámos.

Em *triple buffering*, enquanto o primeiro buffer atualiza o ecrã (o que estiver lá é enviado para o display), um segundo buffer é usado para montar o próximo *frame* que será enviado para o buffer principal (função `page_flip()`). Além disso, um terceiro buffer é utilizado para carregar elementos estáticos, evitando a perda de performance ao carregar sempre os mesmos elementos.

Utilizamos *sprites* para criar os vários elementos visuais do nosso programa cuja forma não é alterada durante a execução do programa, através da função `Sprite *createSprite(xpm_map_t sprite)` e de um XPM previamente definido como, por exemplo, o cursor do rato e os botões do menu. Para além disso, também utilizamos funções para detetar colisões entre os disparos e as personagens que se encontram no ecrã, pela função `check_sprite_collision(Sprite *sp1, Sprite *sp2)`.

Foram usadas as funções 01, 02, 07 com a subfunção 80 do VBE.

A implementação da configuração e funcionalidade referentes ao *video graphics* encontram-se no ficheiro `video_card.c`.

## RTC

O RTC foi configurado de forma a obter a data do dia em que o jogador se encontra. Esta data é representada no menu inicial através do uso de XPM's e *video graphics*.

A implementação da configuração, funcionalidades e interrupções referentes ao RTC encontram-se no ficheiro `rtc.c`.

# Organização do código/estrutura

## **Keyboard Module - 15%**

Este módulo contém as funções desenvolvidas no Lab3 que puderam ser reutilizadas tendo em conta o objetivo do jogo. Estas funções permitem configurar e invocar as interrupções do dispositivo.

## **Mouse Module - 10%**

Este módulo contém as funções desenvolvidas no Lab4 que puderam ser reutilizadas tendo em conta o objetivo do jogo. Estas funções permitem configurar, invocar as interrupções do dispositivo e sincronizar os 3 bytes de informação retornados pelo controlador em cada interrupção.

## **RTC - 5%**

Este módulo contém as funções necessárias para configurar e obter o *Real Time Clock* para depois poder ser colocado no menu inicial.

## **Timer - 10%**

Este módulo contém as funções desenvolvidas no Lab2 das aulas práticas sobre o timer que puderam ser reutilizadas tendo em conta o propósito do nosso jogo. Estas funções permitem fazer a configuração, as interrupções do timer e a sincronização entre os vários dispositivos usados.

## **Video Card Module - 15%**

Este módulo contém as funções desenvolvidas no Lab5 das aulas práticas sobre o *video graphics* que puderam ser reutilizadas tendo em conta o propósito do nosso jogo. Estas funções permitem fazer a configuração do *video graphics* e desenhar algumas das formas que aparecem nos ecrãs, nomeadamente a estrutura mais geral das páginas de desenhar e adivinhar.

## **Utils Module - 3%**

Este módulo contém as funções desenvolvidas no Lab2 das aulas práticas sobre o timer que posteriormente continuaram a ser usadas nos Labs seguintes e por esse motivo puderam ser também reutilizadas no nosso Projeto.

### **Events Module - 12%**

Este módulo faz o tratamento dos *inputs* dados pelo jogador, seja no menu ou dentro do próprio jogo, atualizando os ecrãs e as posições das personagens.

### **Game Auxiliary Module – 15%**

Este módulo contém as funções que fazem a interface do menu e contém as principais funções para as funcionalidades do jogo.

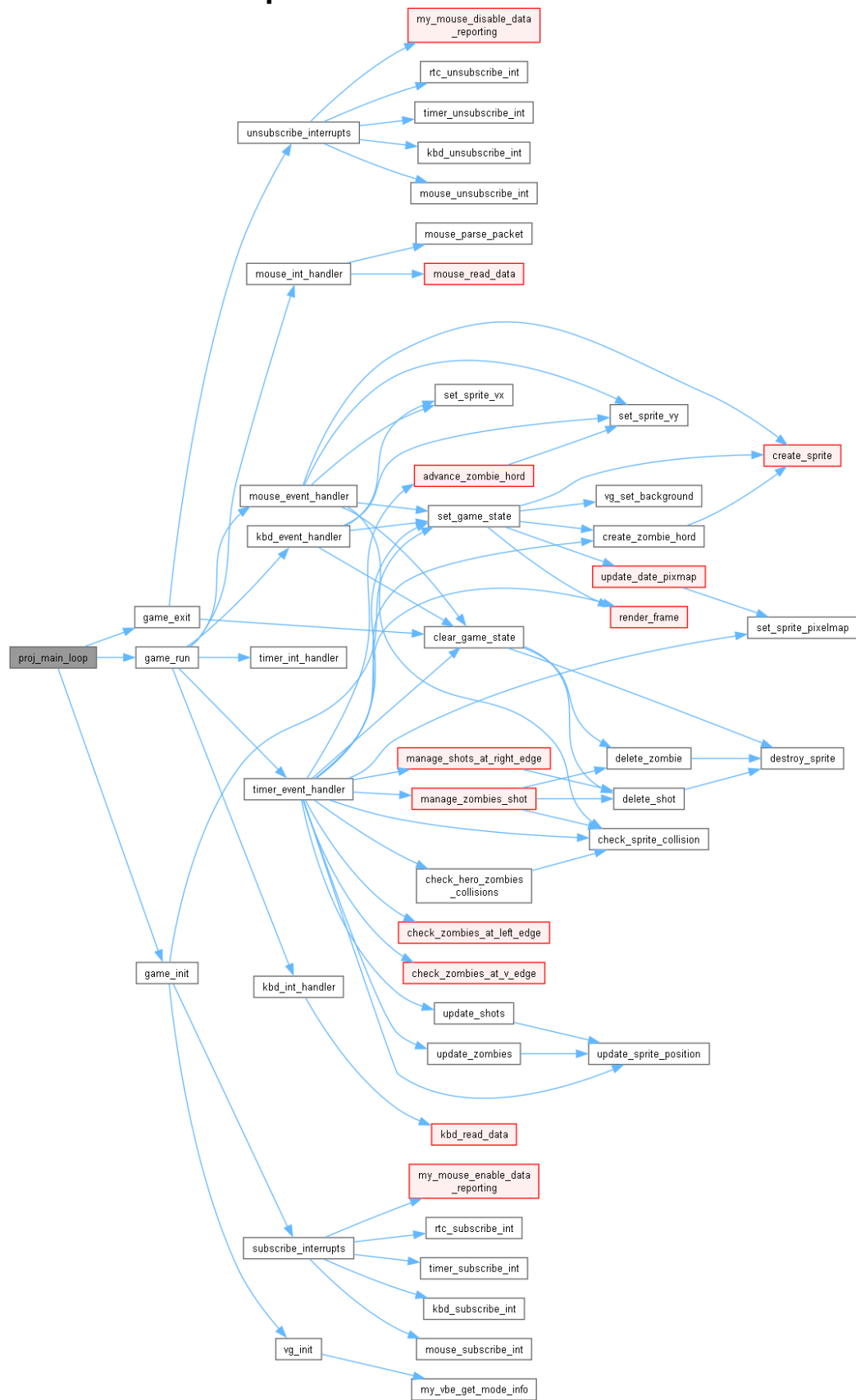
### **Sprites Module - 10%**

Este módulo contém as funções que permitem desenhar as personagens e os pixelmaps necessários.

### **Game Module - 5%**

Este módulo contém o *loop* principal do projeto assim como o *loop* que recebe e trata de cada interrupção.

## Function call Graph



# Conclusões

Neste Projeto conseguimos implementar com sucesso todas as *features* descritas na proposta inicial.

Escolhemos este jogo de forma a ir além dos dispositivos obrigatórios a implementar. Desta forma, foi necessário implementar *Mouse* e *RTC* para abranger grande parte dos dispositivos abordados nas aulas teóricas.

Gostaríamos de ter adicionado as horas, minutos e segundos juntamente com a data de forma a ter mais detalhe, mas, infelizmente, não foi possível completar essa parte.

Foi com trabalho de equipa e com um plano bem elaborado que desenvolvemos todas as *features* essenciais e entregámos em tempo útil.

A execução deste projeto foi benéfica para todos os elementos do grupo, uma vez que fomos capazes de implementar todos os módulos auxiliares pretendidos e explorámos as várias funcionalidades dos múltiplos dispositivos abordados nas aulas.