

# *Crack of Doom*



Turma 6 – Grupo 2

Gonalo Bernardo

[up201606058@fe.up.pt](mailto:up201606058@fe.up.pt)

Joana Ramos

[up201605017@fe.up.pt](mailto:up201605017@fe.up.pt)

# *ÍNDICE*

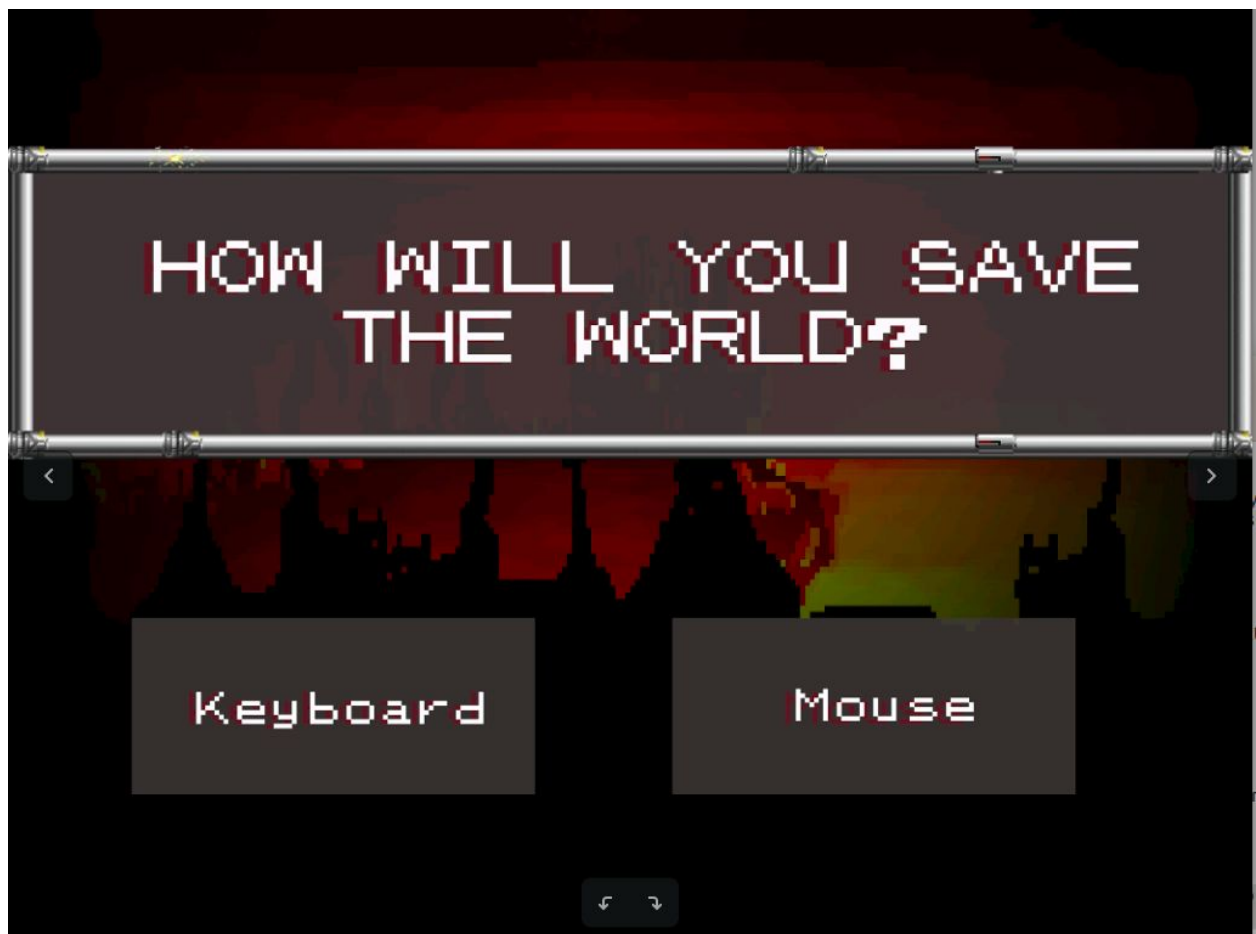
<i>I.</i>	Instruções de Utilização	<i>.p3</i>
<i>II.</i>	Estado do Projeto	<i>.p7</i>
<i>III.</i>	Organização e Estrutura do Código	<i>.p10</i>
<i>IV.</i>	<i>Detalhes da Implementação</i>	<i>.p14</i>
<i>V.</i>	<i>Conclusões</i>	<i>.p15</i>
<i>VI.</i>	<i>Apêndice</i>	<i>.p16</i>

# *I. Instruções de Utilização*

## *ESCOLHA DE PERIFÉRICO*

Ao iniciar-se o jogo, é pedido ao utilizador que escolha como quer salvar o mundo. Utilizando o *keyboard*, ou o *mouse*?

Com o uso do *keyboard*, o rato não tem efeito no jogo e o teclado será usado para ambos os menus como para o jogo em si. Porém, ao verificar-se a opção do rato, o teclado não terá efeito no jogo efetivo, tanto que apenas o mouse controlará a nave espacial principal.



*Figura 1*

## *MAIN MENU*

No momento em que o utilizador inicia o programa, é-lhe mostrado um menu inicial com as seguintes opções de escolha:

- *START GAME*, iniciando um novo jogo.
- *CHANGE CHARACTER*, encaminhando o utilizador para um menu de escolha tanto da nave em uso como da bola de energia em causa.
- *EXIT GAME*, saindo do programa.

Neste menu principal, é possível pressionarem-se a teclas A ou W[cima] e D ou S[baixo] para mover a seta, “SPACE” ou “ENTER” como escolha efetiva, e “ESC” para sair do jogo, **ou então usar-se o rato.**



Figura 2

## *START GAME*

Ao iniciar-se o jogo, há como objetivo destruir todos os drones presentes(inimigos) sem que a bola de energia atinja a área abaixo da nave principal, dentro de um limite de vidas.

O começo efetivo do jogo é feito premindo a tecla “*SPACE*” para lançar a bola. É lançada gerando um ângulo aleatório limitado entre 30° a 150° e movimenta-se colidindo com todas as superfícies dentro da área de jogo bem como com a nave ou os drones. À medida que isto acontece, a bola ganha uma aceleração com o objetivo de dificultar a vitória. A nave espacial principal é movimentada pressionando as teclas A(p/esquerda) ou D(p/direita), caso o keyboard seja selecionado.

Foram discutidos três casos para implementação: pausa, perda de jogo e desistência. A primeira é feita premindo a tecla P e em prática, para a fluidez do jogo até que a mesma tecla seja premida novamente. A perda de jogo é feita quando a bola atinge a grelha abaixo da nave, e redireciona o utilizador para uma mensagem de *Game Over*. Finalmente, a terceira situação é feita dando a liberdade ao jogador de sair quando quiser ao premir o “*ESC*”.

No decorrer do jogo observa-se pela grelha à direita, que o jogo apresenta uma pontuação relativa à destruição dos drones e que vai sendo incrementada.

Figuras 3 e 4



## CHANGE CHARACTER

Aqui, neste menu, o utilizador tem a liberdade de escolher tanto a sua personagem, como a bola de energia que irá ajudá-lo na vitória. Existem três opções para cada um dos objetos referidos. Ora, já está pré-definida uma escolha das três de cada objeto em questão.

A escolha é feita premindo as teclas A ou D para eixo horizontal(p/esquerda ou direita), W ou S para eixo vertical(p/cima ou baixo) e “SPACE” ou “ENTER” para escolha efetiva. Assim sendo, é encaminhado de volta para o Menu Principal.



Figura 5

## II. Estado do Projeto

### DISPOSITIVOS UTILIZADOS

DISPOSITIVO	USO	INTERRUPÇÃO
TIMER	Atualização do estado do jogo	✓
TECLADO	Interface entre user/jogo bem como no menu.	✓
RATO	Interface entre user/jogo e navegação	✓
PLACA GRÁFICA	Desenho de imagens	X

#### TIMER

O uso do timer foi com o objetivo de atualizar o estado do jogo e ecrã, ou seja, através de interrupções do timer desenhar os gráficos do jogo(ambiente, personagem, inimigo, seta) a 60 *frames* por segundo. É dos dispositivos mais importantes por manusear com todo o tipo de informação - *graphics.c* e *game.c*

#### TECLADO

Usado na lógica do jogo, o teclado tem o propósito de criar uma interface entre o utilizador e o programa inteiro. Nos menus descritos, é usado como movimento de escolha de opções. No decorrer do jogo é utilizado para o movimento da barra e o começo do andamento da bola, caso seja escolhido inicialmente. Gere a sua informação conforme o scancode recebido.



## *RATO*

Tal como o teclado, o rato é também utilizado como interface entre o user e o programa. Usado no menu como escolha de opções e utilizado como começo do movimento da bola (“*SPACE*”), o rato efetua uma ligação entre a lógica do jogo e a passagem para tratamento de gráfica. Através de interrupções é feita uma alteração da sua estrutura (movimenta a barra, se for selecionado na escolha do periférico).

## *PLACA GRÁFICA*

A placa gráfica trata-se do periférico mais essencial do projeto. É de notar a sua importância por completar a ligação entre utilizador/lógica/imagem. Assim sendo, trata de desenhar todas as imagens [formato bitmap, com modo vídeo 0x117 de resolução 1024 por 768 píxeis e modo RGB(5:6:5)].

Para que o jogo pudesse apresentar uma fluidez acessível, implementamos a técnica de double buffering e, para que se pudesse observar o andamento do jogo, a atualização do mesmo é feita por interrupções do timer.

A edição de imagens foi feita através dos programas GIMP, Adobe Photoshop e Adobe Illustrator.

# *Organização e Estrutura do Código*

## *BITMAP(7%)*

O módulo de Bitmap está encarregue de carregar, desenhar e apagar imagens em formato Bitmap e tratando das mesmas para que apresentem transparência. Este algoritmo foi desenvolvido e é da autoria de Henrique Ferrolho, publicado no seu blog com o nome de “Difusal”. (<http://difusal.blogspot.pt/2014/09/minixtutorial-8-loading-bmp-images.html>)

Com algumas alterações ao algoritmo e com a assistência do programa Adobe Photoshop, Adobe Illustrator e GIMP. foi criada a possibilidade de desenhar só a imagem que pretendemos.

## *TIMER(8%)*

Código importado do lab2, bem como a possibilidade de obter a configuração do mesmo e ainda a criação de um handler que faz o tratamento das interrupções do timer com base no estado das máquinas de estado implementadas.

## *GRAPHICS(12%)*

Dos módulos mais importantes do projeto. Está responsável por mostrar todas as imagens e fazer o tratamento de quaisquer alterações nas estruturas implementadas.

É este o módulo que inicializa todas as imagens que implementamos, bem como desenhá-las. Para além disso, apresenta algoritmos relativos à movimentação da nave, do ângulo de partida da bola ao iniciar-se, e ainda a criação e alocação de memória dos objetos em questão.

Foram criadas as seguintes estruturas relativas aos aos objetos que, durante o desenvolvimento do jogo, foram surgindo:

- *Explosion\_Arrow arrow*

- Aqui, está contida toda a informação relativa à seta que utilizamos para o menu principal e na escolha de personagens, nomeadamente as coordenadas, o número de opção respetivo e a imagem que a caracteriza.
- *Bar spaceship*
  - Nesta estrutura está contida informação relativa à nave principal que é utilizada no jogo. Tratam-se de informações como as coordenadas iniciais, a sua altura e largura, velocidade e, finalmente, a imagem respetiva(há pré-definição de uma com a possibilidade de mais outras duas).
- *Bar energyball*
  - Trata-se de informação do objeto que apresenta maior movimento, como as coordenadas iniciais, altura/largura(tal como a nave), velocidade e imagem caracterizadora da estrutura. Como já foi referido, é possível alterar-se a imagem para outras duas disponíveis.
- *Drone drones*
  - Estrutura que é alocada através de uma matriz inicial, e na qual se trata das coordenadas do primeiro drone, incrementando a posição através da largura e altura do mesmo. Há apenas uma imagem caracterizadora do objeto em questão.

## GAME(18%)

Sendo que a organização do projeto é um tópico fundamental para o nosso grupo, foram implementadas duas máquinas de estado para facilitar o manuseamento de tanto o jogo, como da personagem. Ora, assim sendo:

- *gameState*
  - *Estados relativos ao tratamento de todas as situações do nosso jogo:*
    - *MAINMENU(menu principal)*
    - *CHOOSE(escolha de opções)*
    - *PREGAME(jogo antes de a bola se movimentar)*
    - *GAME(jogo a partir do momento em que a bola se movimenta)*
    - *GAMEOVER(menu de perda do jogo)*
    - *CHANGE\_CHAR(menu de escolha de personagens)*
    - *END(saída)*
    - *PAUSE(pausa do jogo, e freeze do ambiente)*

■ *VICTORY(menu com o jogo ganho)*

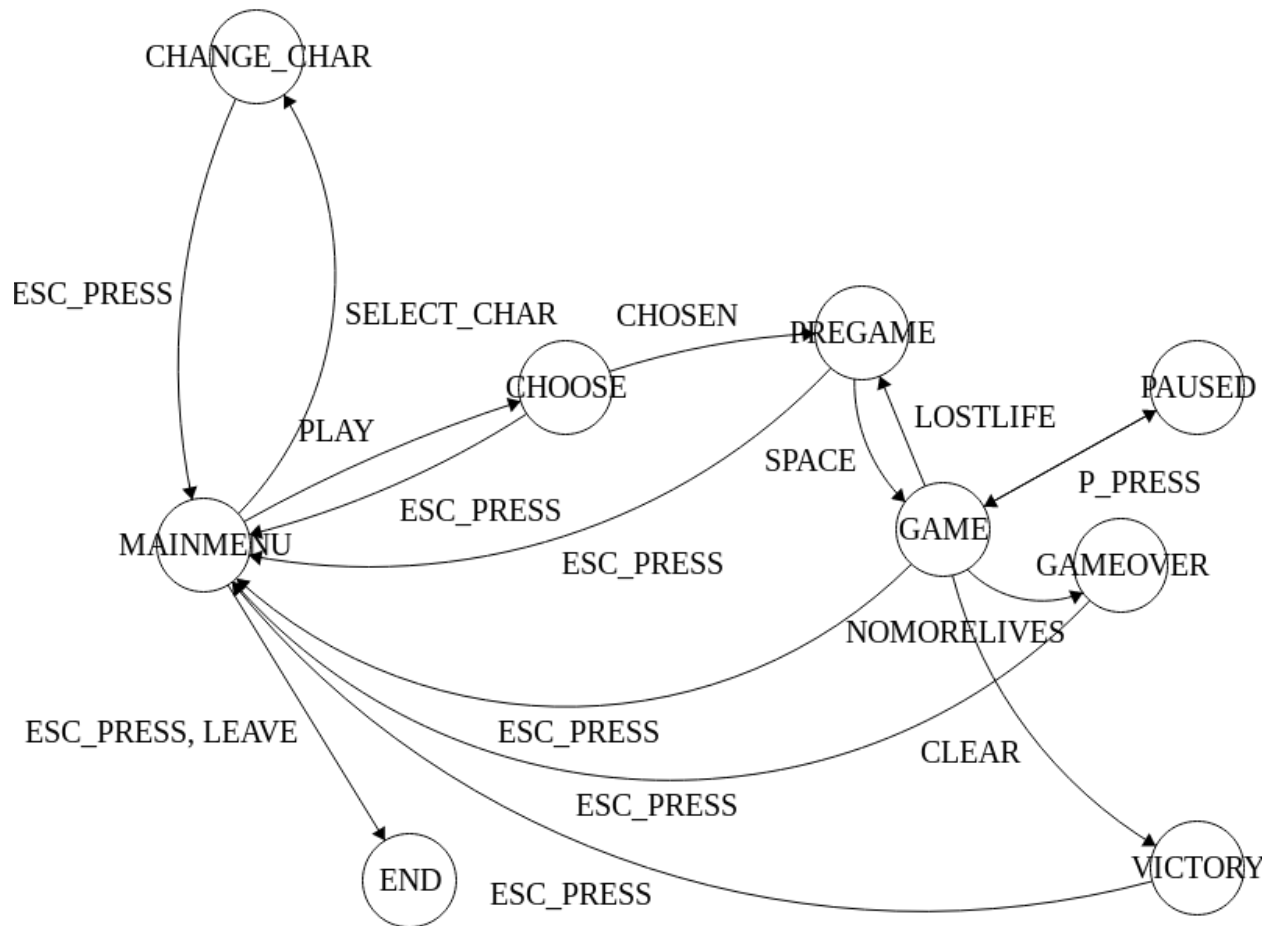


Figura 5

*KEYBOARD(10%)*

Código importado do lab3, bem como a criação de um handler que faz o tratamento das interrupções do keyboard com base no estado das máquinas de estado implementadas.

## *PROJ(6%)*

Tratando-se do módulo principal, é nele que se inicializa o programa. São chamadas, neste módulo, funções como o load de todas as imagens em formato Bitmap; o início do jogo efetivo, e encerramento de todo o programa.

## *VBE(6%)*

Código importado das aulas práticas laboratoriais do laboratório 5.

## *VIDEO\_GR(8%)*

Código importado das aulas práticas laboratoriais do laboratório 5.

## *I8042(5%)*

Código importado das aulas práticas laboratoriais do laboratório 3,4,5.

## *I8254(5%)*

Código importado das aulas práticas laboratoriais do laboratório 2.

## *MOUSE(10%)*

Código importado do lab3, bem como a criação de um handler que faz o tratamento das interrupções do mouse com base no estado das máquinas de estado implementadas.

## *HANDLER(5%)*

Neste módulo implementou-se algoritmos de subscrição aos periféricos, o efeito contrário, e também uma função de `freeMemory()` para desalocar a memória.

## *Detalhes de Implementação*

É relevante mencionar a forma como se implementou a máquina de estados, e a forma como ela organizou o nosso projeto. Facilitou o trabalho de forma para que pudesse correr de forma fluída e bastante consistente.

Queremos ainda dar atenção à forma como implementamos o ângulo aleatório para que a bola pudesse avançar o seu movimento. Será sempre um ângulo entre 30° a 150°, que é convertido para radianos, e excluindo 80° a 100°.

Assim, podemos dizer que quisemos utilizar código com processo de “camada”, para facilitar organização e justificação do uso de código implementado.

# CONCLUSÃO

Em conclusão, sentimos que as aulas práticas laboratoriais tiveram um impacto positivo, bem como a ajuda que nos foi dada pelo professor José Pinto e o nosso monitor. É importante mencionar que houve dificuldades pois enquanto nas práticas trabalhávamos num periférico, nas teóricas a matéria andava muito à frente, não dando muita possibilidade para focar o acompanhamento de ambas as aulas teóricas/práticas. Sentiu-se dificuldades ao fazer a junção de todos os periféricos que usamos e na constante criação de imagens em formato BMP à medida que iam faltando elementos.

Queremos ainda, agradecer ao Henrique Ferrolho e ao seu blog “Difusal”, por ajudar muito na elaboração do projeto.

Finalmente, e em suma, todos os elementos do grupo trabalharam e empenharam-se de forma igual, e, por isso, o trabalho e os objetivos que estávamos à espera estão divididos.

## *APÊNDICE*

*Figura 1 - Menu de escolha do periférico*

*Figura 2 - Menu principal*

*Figura 3 - Jogo efetivo*

*Figura 4 - Perda de jogo*

*Figura 5 - Escolha de personagens*

*Figura 6 - Gráfico de state machine*

*Tabela 1 - Dispositivos utilizados*