

# CaveChaos

Laboratório de Computadores

Mestrado Integrado em Engenharia Informática e Computação

T2-G11:

Luís Miguel Santos Monteiro Saraiva - 201404302

Pedro Filipe Agrela Faria - 201406992

Faculdade de Engenharia da Universidade do Porto  
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

27 de dezembro de 2016

## Resumo

O projeto tem o objetivo de pôr em prática os conceitos lecionados na unidade curricular **Laboratório de Computadores**, com foco na linguagem **C/Assembly**.

Decidimos implementar o jogo CaveChaos que se assemelha ao jogo Flappy Birds.

Neste documento apresentamos, em várias secções, o que consideramos relevante de modo a permitir adquirir um conhecimento global sobre o resultado final do projeto bem como uma reflexão.

A solução que obtemos cumpre parcialmente a maioria dos requisitos do projeto, para além disso, implementamos eficientemente o programa.

A realização deste projeto foi essencial para pôr em prática e compreender melhor os conceitos teóricos inerentes à unidade curricular.

## Conteúdo

1- Instruções de Utilização do Programa .....	4
1.1- Instruções para Iniciar o Programa .....	4
1.2- Instruções para Navegar no Menu.....	4
2- O Jogo .....	5
3- Projeto .....	6
3.1- Dispositivos I/O Utilizados .....	6
3.1.1- Timer .....	6
3.1.2- Teclado.....	6
3.1.3- Rato.....	6
3.1.4- Placa Gráfica .....	7
4- Módulos.....	7
4.1- Módulos .....	7
4.1.1- Timer .....	7
4.1.2- Jogo .....	7
4.1.3- Test3.....	8
4.1.4- Test4.....	8
4.1.5- Test5 e Video_gr .....	8
4.1.6- Sprite.....	8
4.1.7- Highscore .....	8
4.2- Organização do código.....	9
5- Detalhes da Implementação .....	10
6- Distribuição das responsabilidades .....	10
7- Conclusões.....	10

# 1- Instruções de Utilização do Programa

## 1.1- Instruções para Iniciar o Programa

```
Starting daemons: update cron syslogd.
Starting networking: dhcpd nonamed.
Local packages (start): sshd Starting sshd.
done.

Minix Release 3 Version 1.8 (console)

10.0.2.15 login: lcom
Password:

To install additional packages, run 'pkgin'. First do a 'pkgin update'
to update the list of available packages, and then do a 'pkgin' to get
a list of commands. For example, 'pkgin install vim' installs the
'vim' package, and 'pkgin available' will list all available packages.

MINIX 3 supports multiple virtual terminals. Just use ALT+F1, F2, F3
and F4 to navigate among them.

For more information on how to use MINIX 3, see the wiki:
http://wiki.minix3.org.

$ cd Trabalho/src/
$ su
# make
# service run `pwd`/projeto -args "menu $PWD Luis"
```

Para iniciar o programa em modo menu é necessário escrever o código: `service run `pwd`/projeto -args "menu $PWD Luis"`, sendo Luis o nome do jogador. Outra maneira de executar o jogo diretamente é com o comando: `service run `pwd`/projeto -args "start $PWD Luis"`, caso deseje aceder diretamente à pontuação: `service run `pwd`/projeto -args "score $PWD"`.

## 1.2- Instruções para Navegar no Menu

```
===== MENU =====
1- Play
2- View Score
3- Choose character
4- Exit

Please choose an option># _
```

Depois de entrar no menu, são lhe apresentadas 4 opções (*Play*, *View Score*, *Choose character* e *Exit*). Deve carregar no número correspondente à opção que quer para executar, sendo a primeira para jogar, a segunda para mostrar a pontuação dos 5 melhores jogadores, a terceira para escolher o caracter do jogador e por fim, a quarta para sair do jogo.

### 1.3- Instruções para Jogar

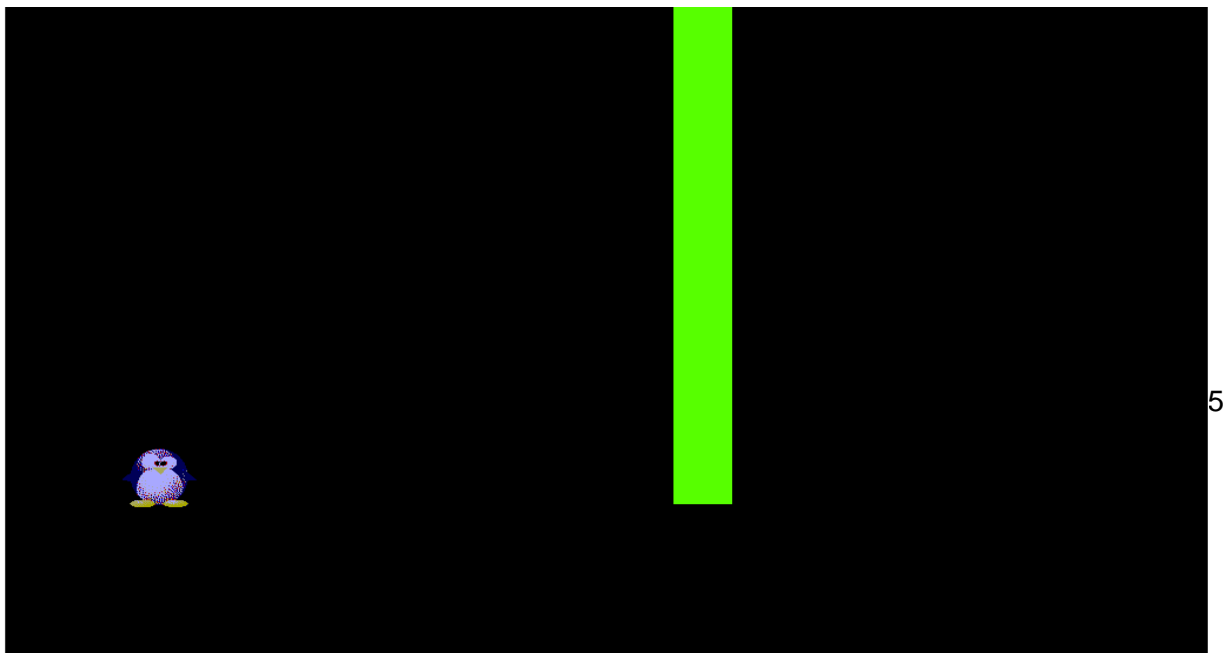
O boneco do jogador encontra-se na parte esquerda do ecrã, podendo mexer-se verticalmente, mas nunca horizontalmente, para poder passar por entre os dois obstáculos.

O boneco do jogador para mover-se, poderá pressionar uma vez ou várias vezes as teclas w e s, para subir e descer, respetivamente, ou movimentar o rato para cima e para baixo. Caso o boneco toque em uma das extremidades dos obstáculos, o mesmo perde o jogo, sendo contado o número de *ticks* em que o jogador se manteve vivo para calcular a pontuação final.

O jogador a qualquer altura do jogo pode carregar no botão P para pausar, e para retomar, clica novamente no botão P. No botão ESC o jogador poderá sair do jogo. Após cada jogo, o resultado é salvo num ficheiro de texto.

## 2- O Jogo

O *CaveChaos* é um jogo baseado no tão conhecido *Flappy Birds* em que o objetivo do mesmo é passar por entre dois obstáculos, sem que o boneco (jogador) toque nas suas extremidades. Pontos vão sendo acumulados à medida que o tempo de jogo avança.



### 3- Projeto

#### 3.1- Dispositivos I/O Utilizados

Dispositivo	Aplicação	Funcionamento
Timer	Responsável pelo controlo do tempo de jogo decorrido	Interrupções
Teclado	Permite terminar e pausar a execução do jogo	Interrupções
Rato	Permite a movimentação do boneco	Interrupções
Placa Gráfica	Responsável pelo ambiente gráfico do jogo	Polling

##### 3.1.1- Timer

O timer funciona por interrupções, a cada interrupção desenha o boneco do jogador, apaga e desenha os obstáculos que aparecem no ecrã, verifica se estes já chegaram ao final do ecrã, verifica se vão acontecer colisões entre os obstáculos e o boneco, incrementa o tempo de jogo, vê se é possível acrescentar mais obstáculos e incrementa a velocidade do jogo a cada 10sec.

Este I/O é utilizado na função start.

##### 3.1.2- Teclado

O teclado funciona por interrupções, é utilizado para navegarmos no menu, para sairmos e pausarmos o jogo e por último para determinarmos se o boneco do jogador se movimenta.

Este I/O foi utilizado nas funções start e getMenuNumber.

##### 3.1.3- Rato

O rato funciona por interrupções, utilizamos o rato para determinar se o boneco se movimenta verticalmente.

Utilizamos este I/O foi utilizado na função start.

### **3.1.4- Placa Gráfica**

A placa gráfica funciona por *polling*, utilizamo-la para desenhar e movimentar o boneco do jogador e os retângulos por quais o boneco tem de passar.

Este I/O foi utilizado nas funções draw\_Player, erase\_Player, erase\_rectangles, draw\_rectangles, check\_collision.

Para além das funções acima indicadas, também consideramos importantes à função change\_heights\_of\_rectangles (que se encontra no módulo jogo), esta função determina uma altura aleatória a um retângulo e através desse valor determinamos a altura do retângulo que está na mesma posição do eixo dos xx, e a função showBestUsersInfo (que se encontra no módulo highscore), esta função permite-nos ordenar todos os utilizadores pela sua pontuação e imprimir os 5 melhores.

## **4- Módulos**

### **4.1- Módulos**

#### **4.1.1- Timer**

Este módulo é responsável pelo *subscribe* e *unsubscribe* dos *interrupts* do timer.

Responsáveis do grupo - Ambos

Peso Relativo – 10%

#### **4.1.2- Jogo**

Este módulo é responsável pelo menu, pelo *display* e movimento do jogador e dos obstáculos, pelo sistema de pontos, pela deteção de colisões e pela mudança de altura dos obstáculos.

Responsáveis do grupo - Ambos

Peso Relativo – 50%

#### **4.1.3- Test3**

Este módulo é responsável pelo *subscribe* e *unsubscribe* dos *interrupts* do teclado.

Responsáveis do grupo - Ambos

Peso Relativo – 10%

#### **4.1.4- Test4**

Este módulo é responsável pelo *subscribe* e *unsubscribe* dos *interrupts* do rato.

Responsáveis do grupo - Ambos

Peso Relativo – 10%

#### **4.1.5- Test5 e Video\_gr**

Estes módulos são responsáveis pela inicialização da parte gráfica.

Responsáveis do grupo - Ambos

Peso Relativo – 10%

#### **4.1.6- Sprite**

Este módulo é responsável pela inicialização dos obstáculos.

Responsáveis do grupo - Ambos

Peso Relativo - 5%

#### **4.1.7- Highscore**

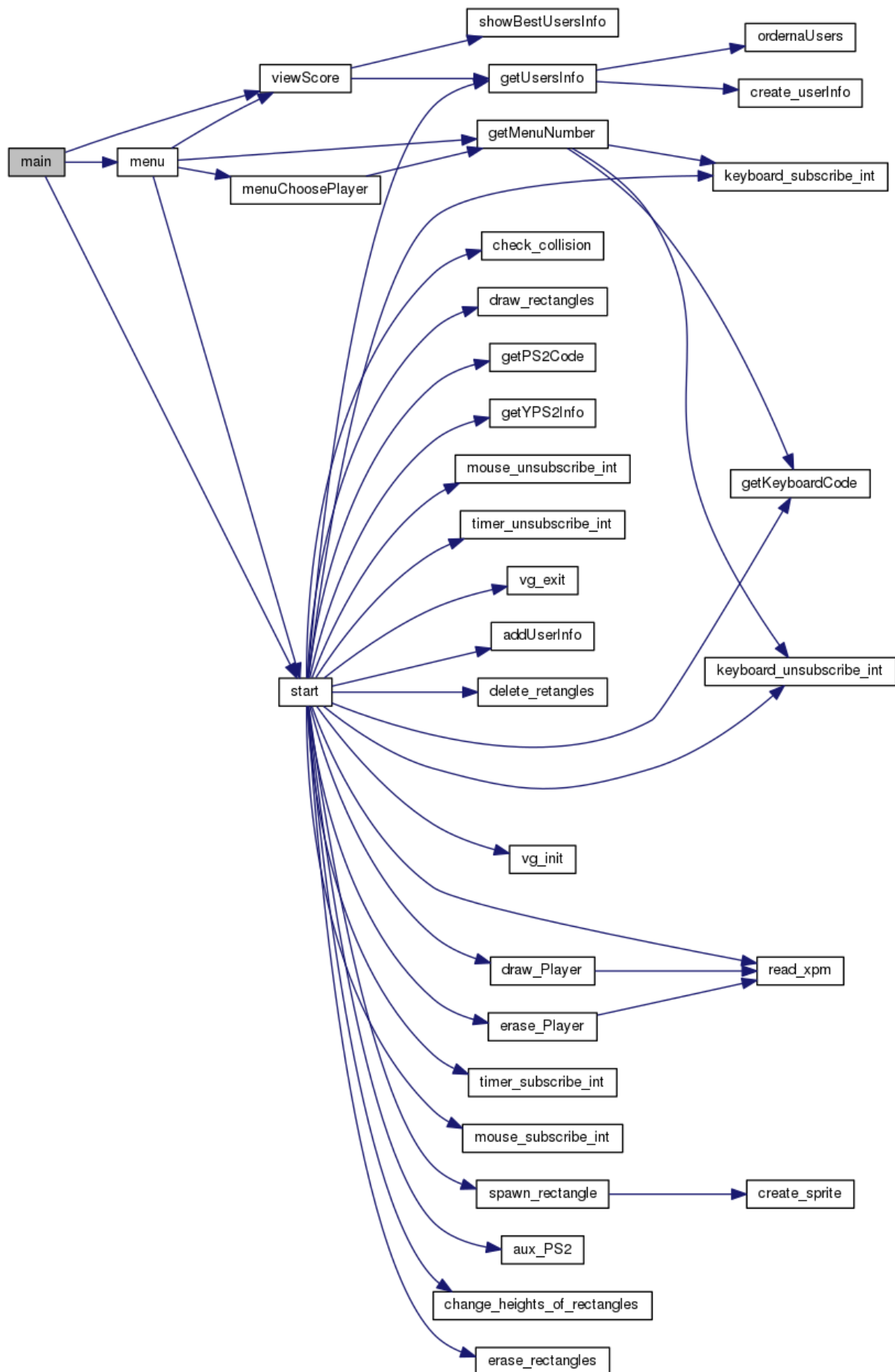
Este módulo é responsável por criar um novo utilizador, adicioná-lo ao ficheiro onde se encontram os outros utilizadores com as suas pontuações, ordenar os utilizadores que se encontram num ficheiro e por imprimir os melhores 5 utilizadores.

Responsável do grupo - Pedro

Peso Relativo – 5%



## 4.2- Organização do código



## 5- Detalhes da Implementação

O domínio da linguagem C foi absolutamente imprescindível para o sucesso da realização de todos os trabalhos propostos ao longo do semestre, incluindo este projeto. A título de exemplo, importa aqui salientar a alocação de memória quando se inicializa os obstáculos e a utilização do tipo de dados *struct*, também, nos obstáculos e nos utilizadores.

## 6- Distribuição das responsabilidades

O projeto foi desenvolvido parcialmente em partes iguais, ambos fizemos a implementação dos seguintes módulos: *Timer*, *Keyboard*(Test3.c), *Rato*(Test4.c), *Placa de vídeo*(Test5.c), *Sprite* e *Jogo*.

Individualmente o Luís Saraiva implementou: Evento do rato (no jogo.c).

Individualmente o Pedro Faria implementou: *Highscores*.

## 7- Conclusões

O grupo utilizou o trabalho da melhor maneira para pôr em prática os conceitos lecionados nas aulas. Trabalhamos desde a primeira hora e todos os prazos foram cumpridos.

A maior dificuldade sentida foi na implementação dos obstáculos e na do Rato.

Concluimos que atingimos quase todos os objetivos que nos tínhamos proposto e que a contribuição foi equitativa e trabalhamos bem como grupo.