

**Where’s Wally?**

Laboratório de Computadores

Relatório Final do Projecto



**Turma 2 Grupo 15:**

Diogo Xavier Ribeiro Pereira (up201305602)

João Henrique Poceiro Vieira de Araújo (up201303962)

**Conteúdo**

[**Breve descrição do projecto** 3](#_Toc439670528)

[**Instruções de instalação** 3](#_Toc439670529)

[**Instruções de utilização** 4](#_Toc439670530)

[**Estado do Projecto** 9](#_Toc439670531)

[**Organização/Estrutura de Código** 10](#_Toc439670532)

[**Diagrama de Chamada de Funções** 12](#_Toc439670533)

[**Detalhes da implementação** 13](#_Toc439670534)

[**Conclusões** 13](#_Toc439670535)

2MIEIC02Grupo 15Página 2 de 13

# **Breve descrição do projecto**

O projecto consiste na criação de um jogo point-and-click, baseado na popular série de livros "Onde Está o Wally?". O objectivo do jogo é encontrar as várias personagens solicitadas que se encontram escondidas em diversas situações ao longo dos mapas disponíveis. Este jogo é bastante intituitivo e apenas requere um bom olho para analisar as paisagens.

# **Instruções de instalação**

Na primeira utilização do programa, deverá percorrer sequencialmente os seguintes passos:

1. Fazer checkout da pasta lcom1516-t2g15
2. Aceder a essa pasta e posteriormente à pasta proj que se encontra no seu interior
3. Adicionar a biblioteca liblm.a à pasta proj/code
4. Copiar o ficheiro proj que se encontra na pasta proj/conf para para etc/system.conf.d
5. Aceder à pasta proj/code e executar a instrução “make”
6. Entrar em modo root através do comando “su”
7. Voltar à pasta proj e executar a instrução “sh run.sh”

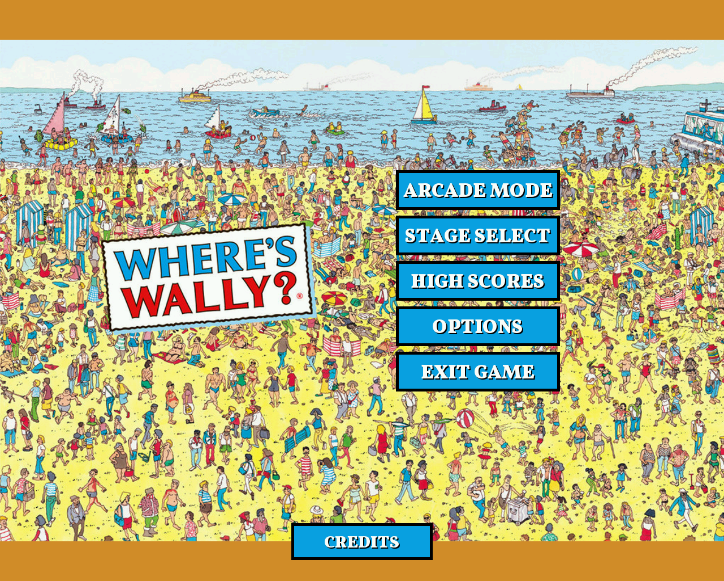
Nota: Se apenas fez checkout da pasta proj ou não o fez para o utilizador lcom, poderá ser necessário alterar o PATH\_RESOURCES que se encontra no ficheiro Utilities.h, na linha 228.

Em execuções futuras apenas será preciso executar o passo 7.

2MIEIC02Grupo 15Página 3 de 13

# **Instruções de utilização**

1. **Menu Principal**

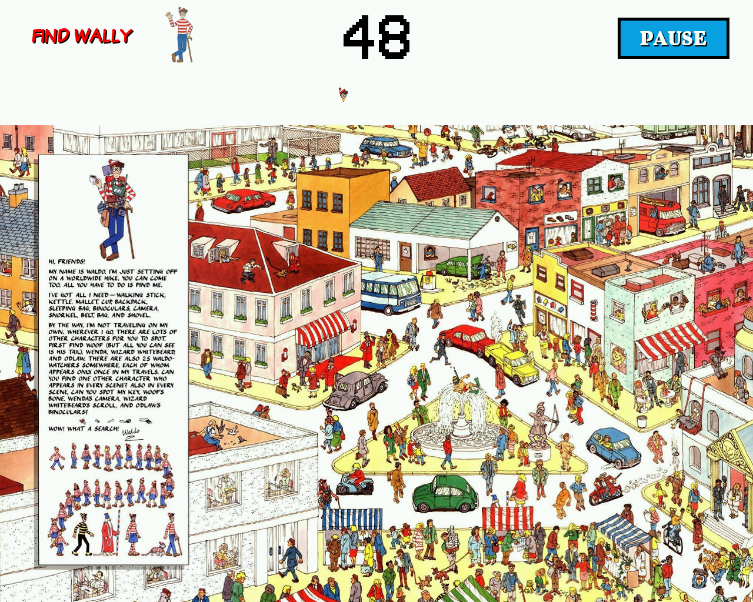


Este é o primeiro menu que o utilizador vê ao iniciar o programa.

O utilizador pode então escolher uma das várias opções, usando ou o rato ou

o teclado (teclas W,A,S,D,Enter e ESC).

1. **Modo Arcade**



Ao seleccionar o modo Arcade no Menu Principal, o jogador é então levado para

o primeiro mapa de muitos. Cabe então a tarefa de descobrir o Wally algures na imagem. Quando o descobrir, necessita de carregar no botão esquerdo do rato para o seleccionar. Cada nível tem um tempo limite de 60 segundos, após os quais, caso não consiga encontrar o Wally, o modo Arcade termina e o utilizador retorna ao menu Inicial. Carregando no botão PAUSE ou na tecla ESC, o jogador pode pausar o nível.

2MIEIC02Grupo 15Página 4 de 13

O princípio é sempre o mesmo e prossegue até o jogador conseguir completar todos os mapas. Caso o faça, pode então registar uma pontuação máxima, sendo solicitada de seguida a introdução do seu nome.



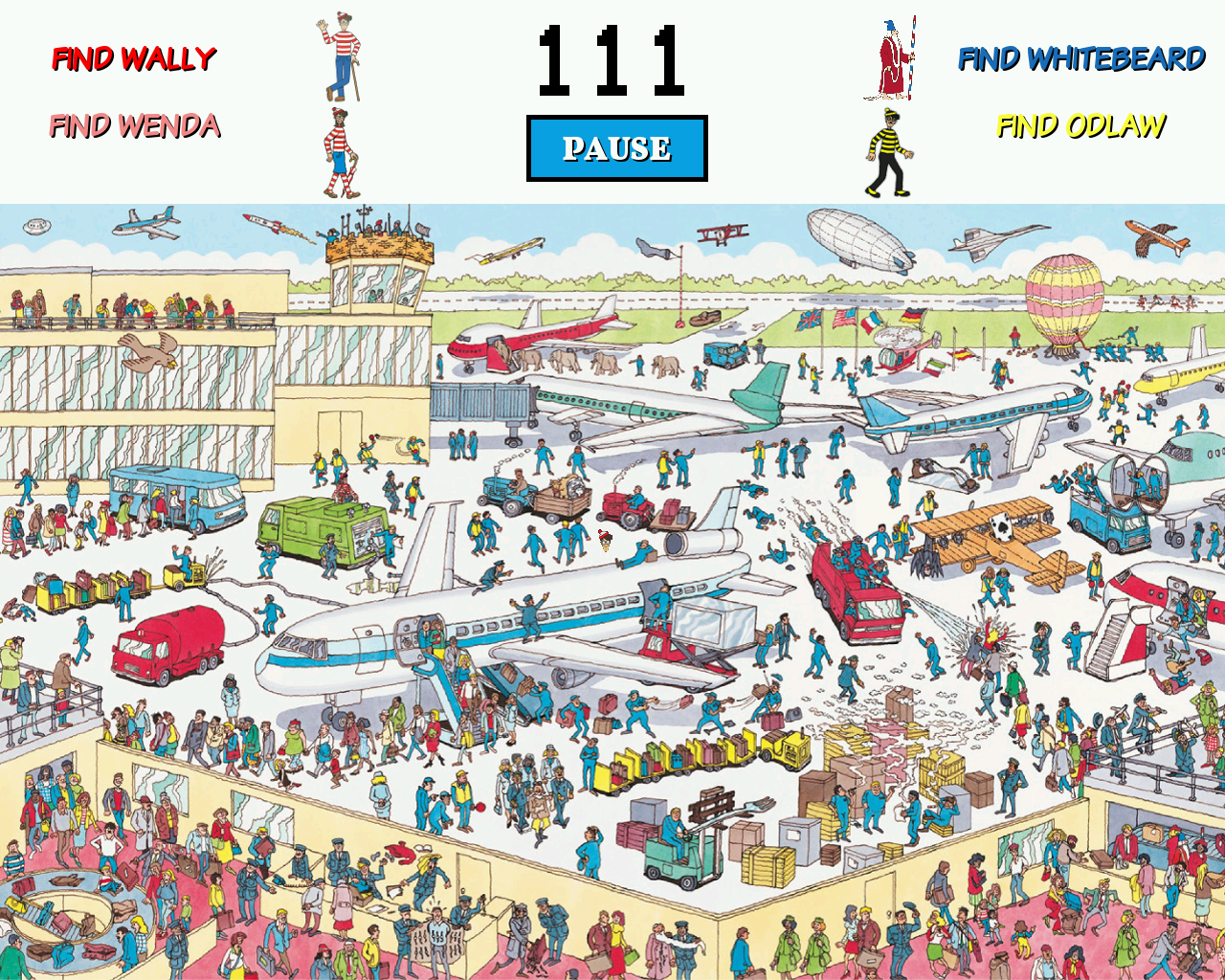
1. **Modo Escolha de Mapa**



Existem ao todo 9 mapas por onde escolher.

Para fornecer alguma longevidade e diversão ao jogo, ao contrário do modo Arcade em que apenas é necessário descobrir a posição do Wally, neste modo junta-se a tarefa de descobrir os três companheiros de viagem do Wally, cada um com características diferentes. Devido à dificuldade acrescida, o tempo limite é de 120 segundos.

2MIEIC02Grupo 15Página 5 de 13

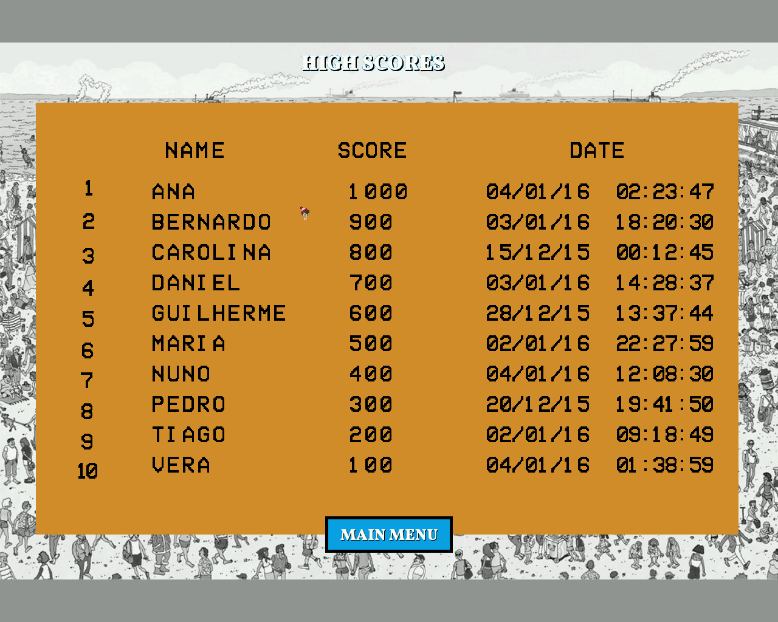


Terminando o tempo ou descobrindo os 4 elementos, o jogador é redirecionado para o menu de Escolha de Mapa, podendo escolher outro mapa ou voltar ao menu principal.

1. **Visualizar Pontuações**

Se quiser verificar as pontuações obtidas no modo Arcade, basta selecionar o menu “High Scores”, que fará a listagem das dez melhores pontuações.

Estas pontuações são tanto maiores quanto menor o tempo que o jogador demorar em cada nível.

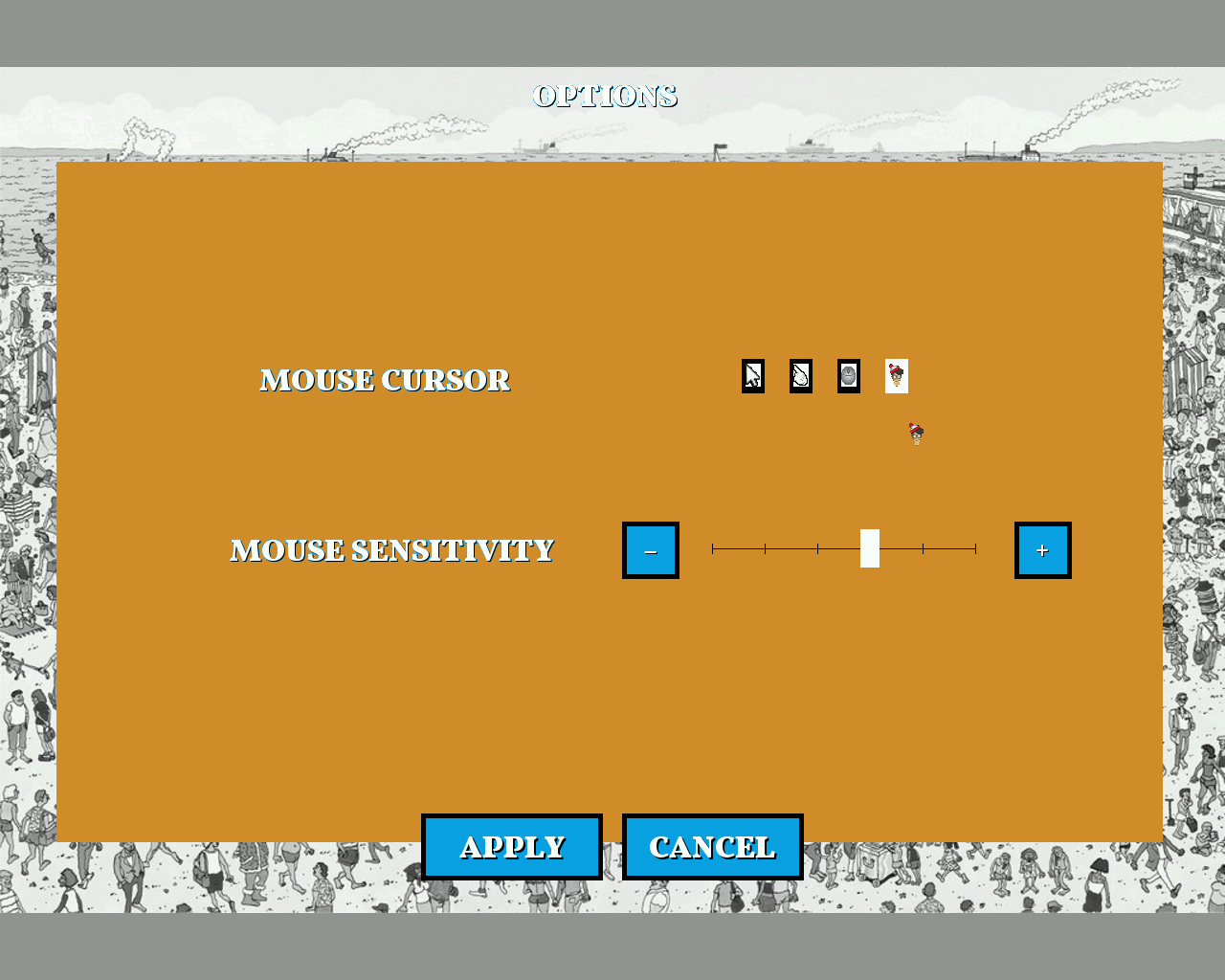
****

2MIEIC02Grupo 15Página 6 de 13

1. **Menu de Opções**

Como sabemos que cada jogador tem as suas preferências, oferecemos a

possibilidade de customizar o cursor à maneira de cada um. Neste menu o utilizador pode seleccionar um dos vários cursores disponíveis e alterar o nível de sensibilidade do movimento do rato.



Seleccionando o botão “Apply” irá aplicar as alterações selecionadas pelo utilizador e retorna ao menu Principal. Por outro lado, o botão “Cancel” apenas retorna ao menu Principal, sem alterar as definições.

1. **Sair do Jogo**

Este botão é bastante explicativo. Tal como o nome indica, quando o utilizador

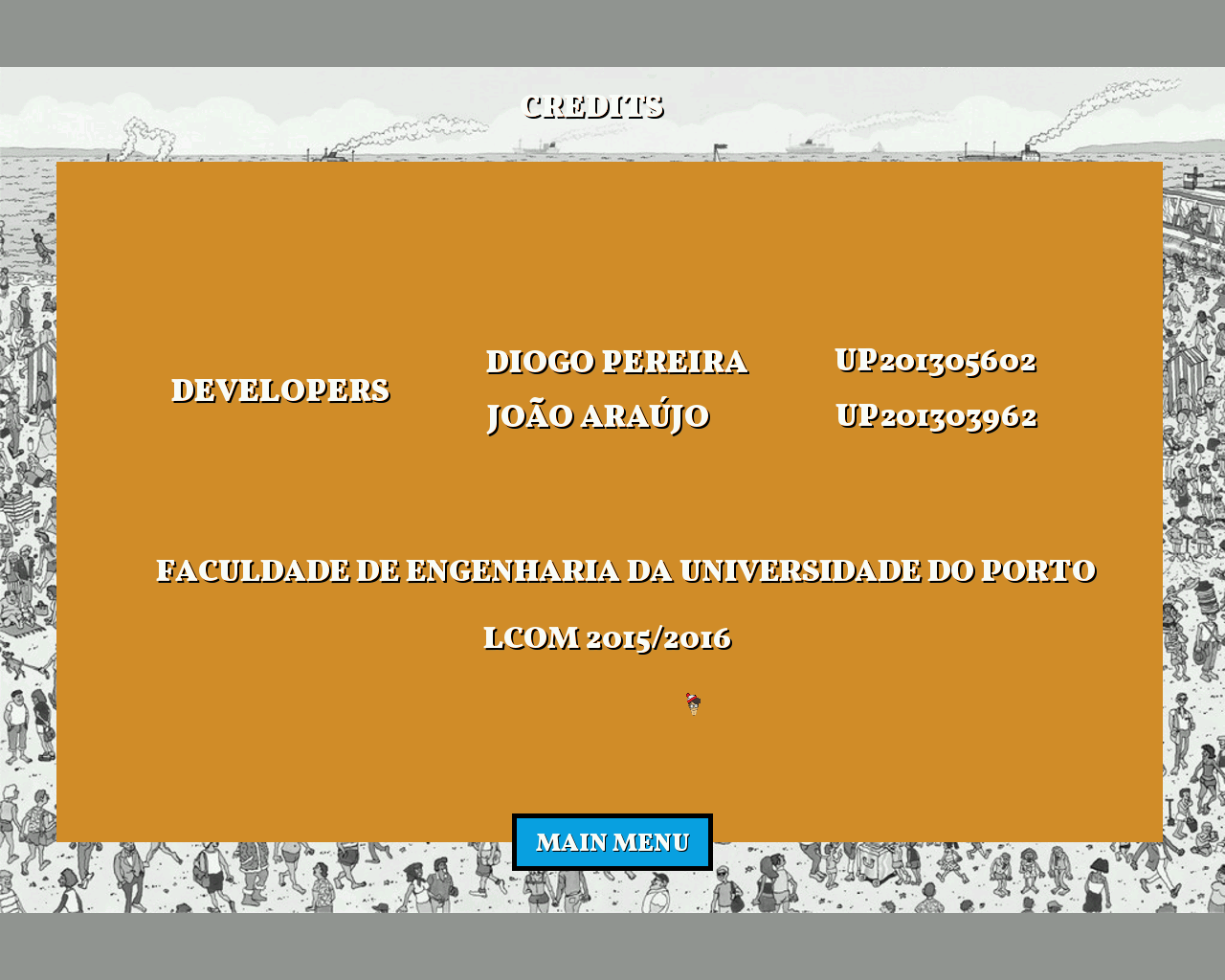
quiser terminar a aplicação, apenas necessita de seleccionar este botão.

2MIEIC02Grupo 15Página 7 de 13

1. **Créditos**

Por fim, se estiver interessado, este menu indica quem foram os

desenvolvedores e o âmbito em que este projecto foi concebido.



2MIEIC02Grupo 15Página 8 de 13

# **Estado do Projecto**

|  |  |  |
| --- | --- | --- |
| **Dispositivo** | **Função** | **Interrupções/*polling*** |
| Timer | Controlar a *frame-rate* e fazer a contagem do tempo limite | Interrupções |
| Teclado | Navegar nos menus e inserir o nome do jogador se este obtiver um highscore | Interrupções |
| Rato | Navegar nos menus e procurar as personagens | Interrupções |
| Placa Gráfica | Desenhar os vários estados de jogo | Não |
| RTC | Obter a data e hora para as pontuações | *Polling* |

**Timer:**

As interrupções do Timer são úteis para gerir a *frame rate* do jogo, estando este dispositivo responsável por definir os *frames-per-second*. Por outro lado, o tempo restante para cada nível é actualizado também por estas interrupções (*WallyNIX.c*, entre as linhas 54 e 64).

**Teclado:**

As interrupções do Teclado servem para navegar nos menus e, quando o utilizador obter uma das melhores pontuações, fornece as letras que compõem o nome desejado (*WallyNIX.c*, entre as linhas 49 e 52).

**Rato:**

É o principal periférico do jogo. Através das suas interrupções, actualiza a posição actual do cursor e os botões do rato (*WallyNIX.c*, entre as linhas 65 e 68).

Existem duas estruturas de dados importantes para a sua utilização: *Position* e *Button*. A primeira guarda as coordenadas (x,y) tanto da posição actual do rato como das personagens a encontrar no jogo, auxiliando facilmente na selecção destas. A segunda serve para criar os vários botões dispersos pelos menus, sendo que cada um possui duas *Position*, uma para o canto superior esquerdo e outro para o canto inferior direito. Através da comparação destas com a *Position* do rato é possível gerir a selecção do rato.

**Placa Gráfica:**

É nela que vai ser desenhado tudo aquilo que é visualizado. O modo de vídeo escolhido foi o modo VESA 16 bit com resolução 1280x1080. São frequentemente desenhados *Bitmaps* e imagens *Font*.

2MIEIC02Grupo 15Página 9 de 13

É utilizado o método de *triple buffering*, ou seja, para além do buffer da placa gráfica que é visualizado, existe também um segundo buffer para onde é desenhado tudo antes de ser copiado para o buffer principal e um terceiro buffer dedicado exclusivamente a um desenho mais fluido do rato.

**RTC:**

É o único dispositivo que funciona em modo *polling*. Este dispositivo apenas é solicitado quando se pretende retornar a data e hora actuais, para guardar numa pontuação.

# **Organização/Estrutura de Código**

**Contribuição:**

A gestão de partes aproximou-se da planeada na especifição, sendo a contribuição total de cada elemento mais ou menos equitativa.

Por serem módulos extensos, os menus *Arcade*, *StageSelect* e *HighScores* foram feitos em conjunto, dividindo tarefas consoante a necessidade. Couberam então ao elemento Diogo Pereira os módulos *Timer*, *Mouse* e menu *Options*, enquanto que o elemento João Araújo ficou responsável pelos módulos *Keyboard*, *Graphics* e *RTC*.

***Bitmap.c* – 5%**

Contém a estrutura Bitmap e as respectivas funções do constructor, desenho e destructor (baseadas no código disponível na seguinte hiperligação <http://forums.fedoraforum.org/archive/index.php/t-171389.html>).

Através de ficheiros .bmp é possível desenhar várias imagens.

***Game.c – 30%***

Contém estruturas para todos os menus, botões e mapas. É também aqui que estão definidas as funções de constructor, desenho, actualização e destructor para cada uma delas.

***Graphics.c – 10%***

Contém as funções para entrar e sair do modo gráfico e desenhar as várias formas utilizadas.

***Keyboard.c – 5%***

Contém as funções para subscrição de interrupções do teclado e leitura do scancode.

2MIEIC02Grupo 15Página 10 de 13

***Main.c – 5%***

Inicia o LOG (útil para a descoberta de erros no desenvolvimento), chama a função para entrar no modo gráfico e cria um objecto do tipo *WallyNIX* que vai ficar em loop num ciclo até o utilizador pretender desligar.

***Mouse.c – 10%***

Contém uma estrutura *Mouse* que guarda a posição e botões do rato. Tem também as funções para a subscrição de interrupções do rato e leitura e actualização dos dados do mesmo.

***Postion.c – 2%***

Contém a estrutura *Position* que representa uma coordenada (x,y). Para além do constructor e destructor contém uma função que calcula a distância entre duas *Position*.

***RTC.c – 5%***

Contém as funções de subscrição do RTC e leitura da data e hora actual.

***Timer.c – 5%***

Contém a estrutura *Timer* que guarda um valor que é constantemente incrementado. Tem também as funções de subscrição e actualização do contador.

***Utilities.c – 5%***

Diversas funções e macros úteis para todo o projecto, funções estas que podem ser tão diversas como arredondar um número ou receber o caminho para uma imagem.

***VBE.c – 3%***

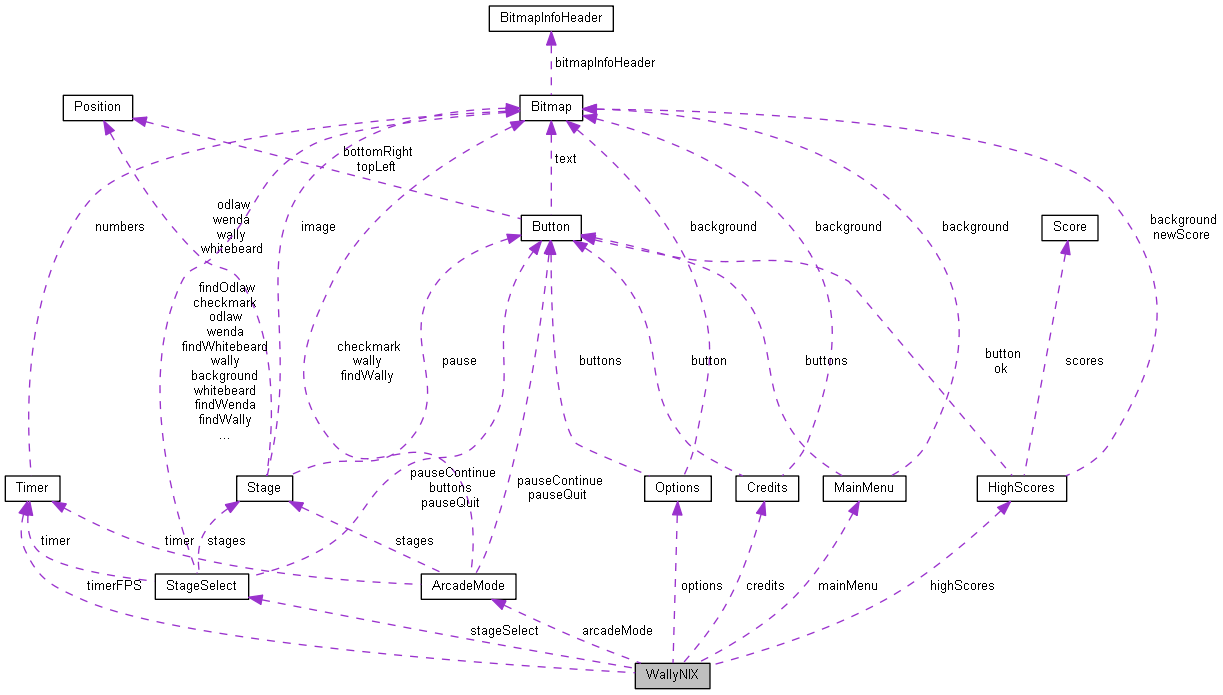
Contém funções que nos dão as características do modo gráfico.

***WallyNIX.c – 15%***

Contém uma estrutura *WallyNIX* que funciona como aplicação principal. Tem também as funções de actualização de interrupções, *inputs* e menus, bem como chama as respectivas funções de desenho.

2MIEIC02Grupo 15Página 11 de 13

# **Diagrama de Chamada de Funções**



A função driver\_receive() é chamada em *WallyNIX.c*, na linha 42, na função que gera as interrupções do timer, teclado e rato. Isto ocorre sempre que o objecto WallyNIX é actualizado.

2MIEIC02Grupo 15Página 12 de 13

# **Detalhes da implementação**

A maior parte das funções relativas aos periféricos usados já tinham sido desenvolvidas nas aulas laboratoriais (com excepção do RTC) por isso não houve grandes problemas nesse aspecto. Talvez o mais trabalhoso foi corrigir uns erros na função responsável por actualizar o rato, o que atrasou um bocado o projecto em relação ao plano estipulado.

A utilização de bitmaps, como não foi abordada nas aulas laboratoriais, não foi tão evidente no início, mas depressa se implementou com a ajuda dos monitores.

A utilização de máquinas de estado foi bastante intuitiva e permitiu uma fácil implementação dos vários menus. A aula teórica dada pelo monitor Henrique Ferrolho ajudou bastante a perceber este conceito.

Por outro lado, e como foi a primeira vez em contacto com a linguagem C, e vindo com bastantes hábitos de C++, a utilização do conceito de objectos e até mesmo a diferença entre char\* e strings dificultou bastante a fase inicial.

# **Conclusões**

Esta é daquelas unidades curriculares onde se se perceber tudo bem, é bastante intuitivo e interessante, mas às vezes faltando perceber um pequeno detalhe o resto do trabalho fica muito dificultado. Talvez uma pequena introdução a C no início do semestre (visto que é o primeiro contacto para a maioria dos alunos) tivesse ajudado em alguns aspectos.

O svn da máquina virtual do minix não é muito fiável e na primeira parte do semestre tivemos imensos problemas por causa disso, fosse por simplesmente não funcionar com a internet da FEUP ou por ser impossível às vezes transferir imagens directamente para o Eclipse, penalizando-nos em um ou dois labs por não termos conseguido submeter tudo a tempo. No entanto, após termos aliado ao svn existente um cliente externo de svn (neste caso TortoiseSVN) ambos estes problemas resolveram-se.

Um aspecto a frisar é que não podemos deixar de sentir que o trabalho exigido não corresponde de todo à atribuição de 6 créditos, pelo menos comparativamente com outras cadeiras com ECTS iguais ou superiores.

Para não haver só pontos negativos, chegando ao final do semestre temos realmente a noção de que aprendemos bastante e é gratificante ver o projecto final desenvolvido depois de tanto esforço.

2MIEIC02Grupo 15Página 13 de 13