U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

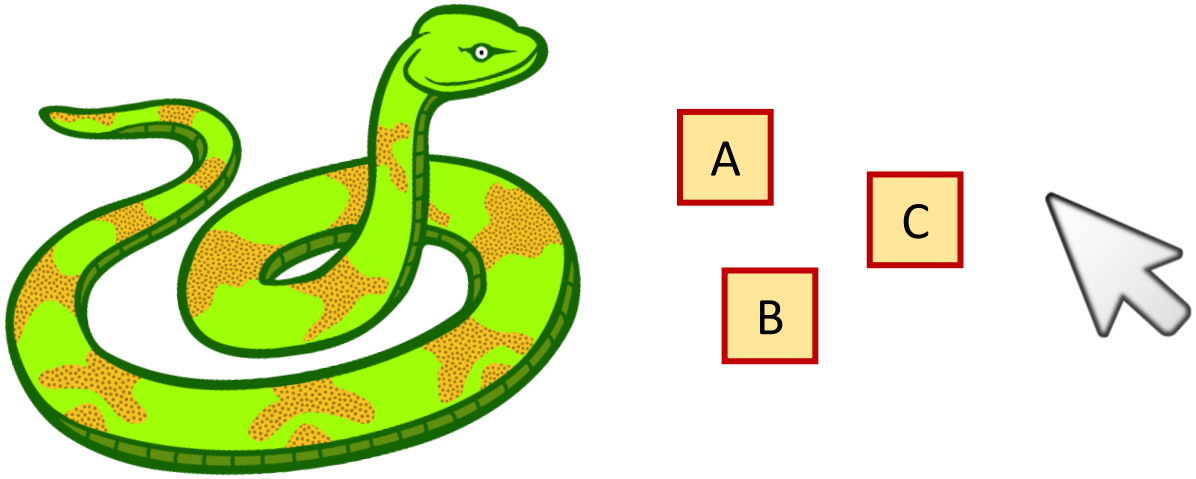
Snaktionary

("Dictionary Snake")

Laboratório de Computadores

Turma1_G09

1 de Janeiro de 2018



Bruno André de Oliveira Vale Marques Fernandes - up200707284@fe.up.pt

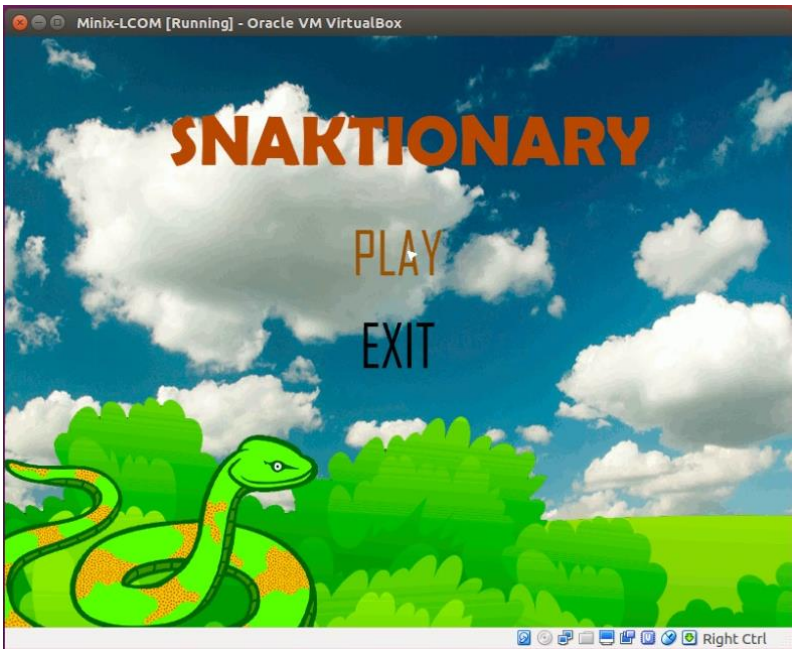
Sandro Miguel Tavares Campos - up201605947@fe.up.pt

ÍNDICE

Instruções ao utilizador

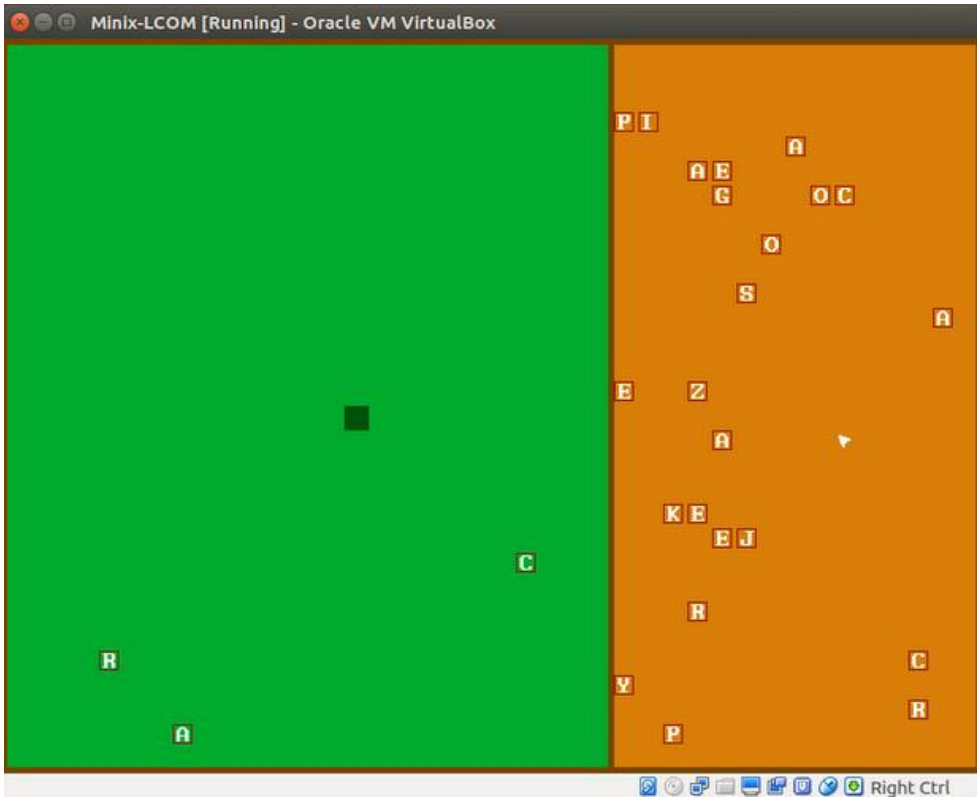
Menu principal

O menu principal é a primeira interação que o utilizador tem com a aplicação. Neste são-lhe apresentadas 2 opções, "Play" e "Exit", que podem facilmente ser selecionadas usando o rato, permitindo aceder ao cenário de jogo e sair da aplicação, respetivamente. Estas interações podem ser observadas nas imagens abaixo.



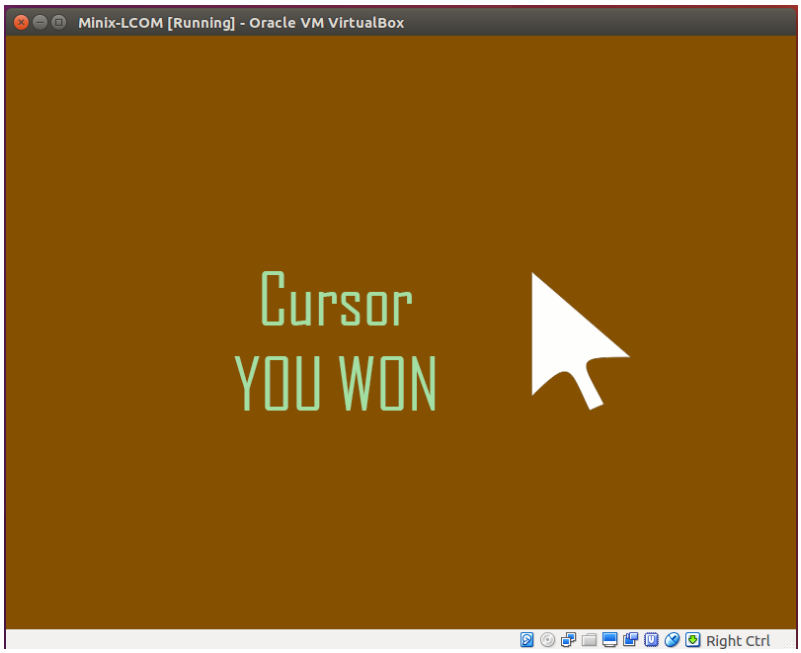
Jogo

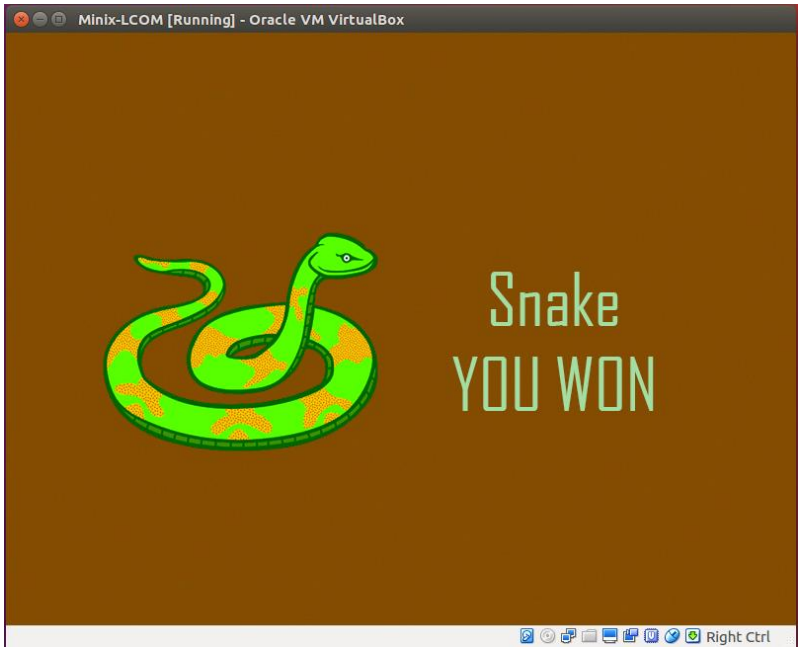
"Snaktionary" é um jogo *multiplayer*, na qual ambos os jogadores têm o mesmo objetivo, completar as palavras que aparecem no ecrã fazendo colidir o seu personagem com as diversas letras que as compõem, e que estão dispersas pelo mesmo. As palavras aparecem uma após outra, sendo que para a cobra aparecem unicamente as letras que compõem a palavra, enquanto que para o cursor aparecem além destas, outras, que dificultam o trabalho de pesquisa. Cada jogador encontra-se por isso a cargo do controlo de um periférico, o teclado associado à cobra e o rato associado ao cursor, na sua parte respetiva do ecrã de jogo. Na seguinte imagem podemos observar o início de um jogo com a palavra "CAR".



Os controlos para a cobra são as teclas W, A, S e D, e para o cursor é necessário o uso do movimento e o do botão esquerdo do rato, para seleção das letras.

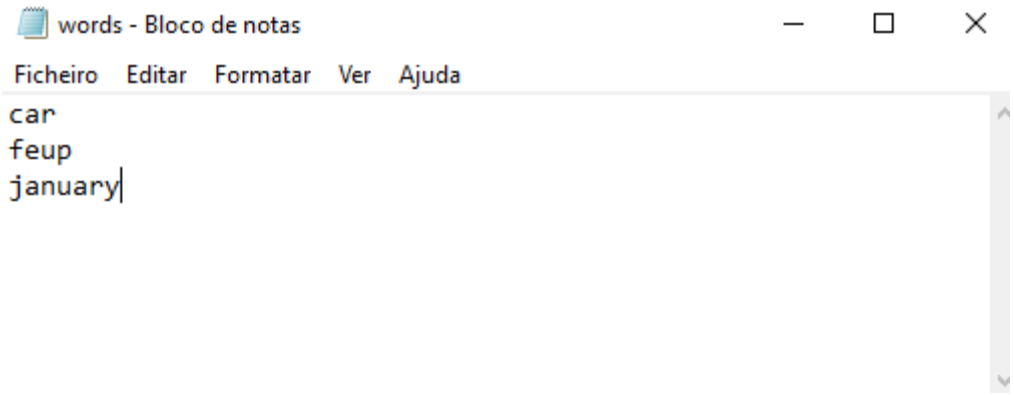
O jogador declarado vencedor é aquele que, ou consegue completar primeiro a sequência de palavras, ou cujo oponente cometa um erro de jogo. Um erro comum a ambos os jogadores é a seleção de uma letra fora da sequência correta, i.e., por exemplo "CR" em vez de "CA" para a palavra "CAR". Além disso, no caso da cobra, a colisão com o seu próprio corpo ou com os limites disponíveis do seu ecrã também a desqualifica, ganhando o seu oponente, o cursor. No fim do jogo é apresentado no ecrã o vencedor e retorna-se ao menu. Seguem-se 2 figuras ilustrando os ecrãs para os vencedores.





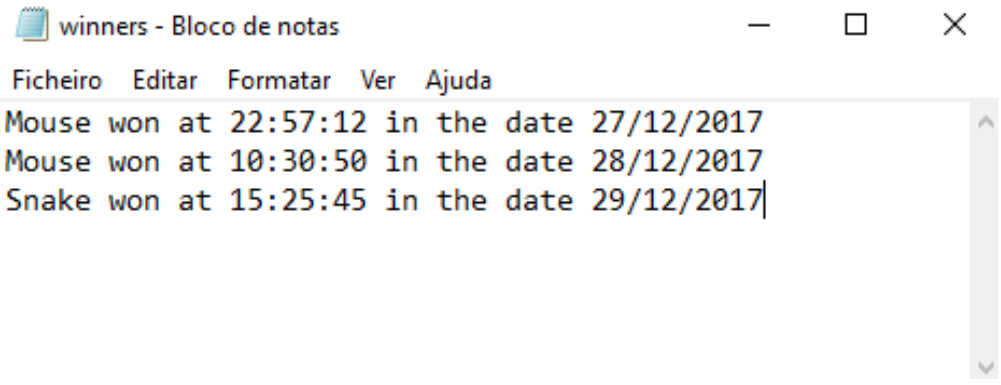
Dicionário de palavras

O ficheiro "words.txt" presente no diretório "res" da aplicação é uma listagem de palavras a serem apresentadas em jogo num determinado momento. O utilizador pode alterar o seu conteúdo se assim o quiser, tendo em atenção que deve colocar apenas uma palavra por linha, e o seu número não pode exceder o valor 20 (valor predefinido aquando da codificação do programa). Inicialmente o ficheiro já estará preenchido de forma a que o utilizador possa jogar prontamente. No exemplo que se segue, as palavras a serem jogadas seriam "CAR", "FEUP" e "JANUARY", uma após outra.



Historial de vencedores

O ficheiro "winners.txt" regista o historial de vencedores do jogo, assim como a hora e data em que esse jogo terminou. O formato deste tipo de informação aparece como se observa na imagem seguinte.



Na instalação do programa, o ficheiro "words.txt" é copiado para um diretório mais conveniente pelo que, quando alterado, deve proceder-se a uma nova instalação, como vai ser esclarecido mais adiante. É também neste diretório que o ficheiro winners.txt vai ser criado.

Estado do projeto

O jogo "Snaktionary" faz uso de diferentes dispositivos de entrada / saída. Seguem-se as descrições de cada utilização.

Timer

O timer tem como principal função a atualização do estado do jogo, num determinado momento. Assim sendo, através de uma variável inicializada a zero e que é incrementada a cada interrupção do timer, conseguimos controlar diferentes aspetos do jogo sendo o principal o número de *frames* por segundo em que este é jogado. Quanto maior for o número de *frames* por segundo, maior é a velocidade com que a cobra se desloca, e por uma questão de simplicidade e eficácia, este foi fixado no valor de 15 *frames* por segundo.

O timer é usado com interrupções nas funções de menu e de jogo presentes no ficheiro *game.c*, respetivamente, *main_menu* e *play_game*, e a sua configuração encontra-se no ficheiro importado das aulas laboratoriais do timer, *timer.c*.

Teclado

O teclado tem como função controlar a cobra no modo de jogo, através do envio de *scancodes* para o *KBC*. Os seus controlos, já apresentados anteriormente, são as teclas W, A, S e D (W faz com que a cobra se desloque na direção Norte, A na direção Oeste, S na direção Sul e D na direção Este).

Este dispositivo, usado com interrupções também nas funções *main_menu* e *play_game* do ficheiro *game.c*, possui a sua configuração nos ficheiros *kbd.h*, *kbd.c* e kbd_asm.S, todos resultantes de aulas laboratoriais anteriores, com as devidas e necessárias alterações.

Rato

O rato é um dispositivo muito importante na aplicação, sendo juntamente com a placa gráfica, aquele que é mais utilizado. Não só é integrante no menu, onde é necessário para selecionar os botões, com um click no seu botão esquerdo, como é um

dos periféricos usados dentro do jogo para controlar o cursor, utilizando, portanto, também o seu movimento.

Recorrendo a interrupções, este dispositivo é usado nas funções de menu e de jogo, *main_menu* e *play_game* no ficheiro *game.c*, e possui a sua configuração nos ficheiros *mouse*.h e *mouse*.c, também resultantes de aulas laboratoriais anteriores. A sua posição é atualizada recorrendo à função *update_cursor*, importante para outras funções como *menu_handling* e *test_collision_cursor*, ambas no ficheiro *game.c*.

Placa gráfica

A placa gráfica é o dispositivo responsável pelo desenho de imagens no ecrã através da colorização de pixéis. O modo de vídeo utilizado neste projeto, 0x115 de resolução 800x600, recorre a cores RGB[8:8:8] – modo de cor direta de 24 bits. Para se iniciar este modo de vídeo recorre-se ao uso da função *vbe_get_mode_info*, também já implementada nas aulas laboratoriais da placa gráfica.

As imagens usadas são de formato PNG, entre elas a imagem que possui a fonte de letra usada para escrever as palavras do jogo, *font.png*. Na obtenção de um carater em particular a partir da imagem fonte, elaborou-se um algoritmo, *vg_tile*, presente no ficheiro *video_gr.c*.

Para permitir uma maior fluidez do programa e facilitando a sobreposição de imagens, implementou-se a técnica de *double buffering*. Podemos encontrar as funções associadas a esta no ficheiro *video_gr.c*, nomeadamente as funções *vg_copy* e *vg_free* que permitem que os dois *buffers* gráficos interajam fazendo a troca de *buffers*, e a última libertando o *buffer* auxiliar.

RTC

O RTC (*Real Time Clock*) é usado para obter a hora e data do computador num determinado momento. Esse momento é, na nossa aplicação, toda a vez que o jogo termina e é anunciado um vencedor, sendo este posteriormente adicionado a um ficheiro de texto criado para o efeito – "winners.txt". Com isto é possível gravar-se informação acerca do historial de jogo. As funções inerentes às suas funcionalidades, *rtc_time* e *rtc_date*, encontram-se presentes no ficheiro *rtc.c*.

Em seguida apresenta-se uma tabela com um sumário do apresentado:

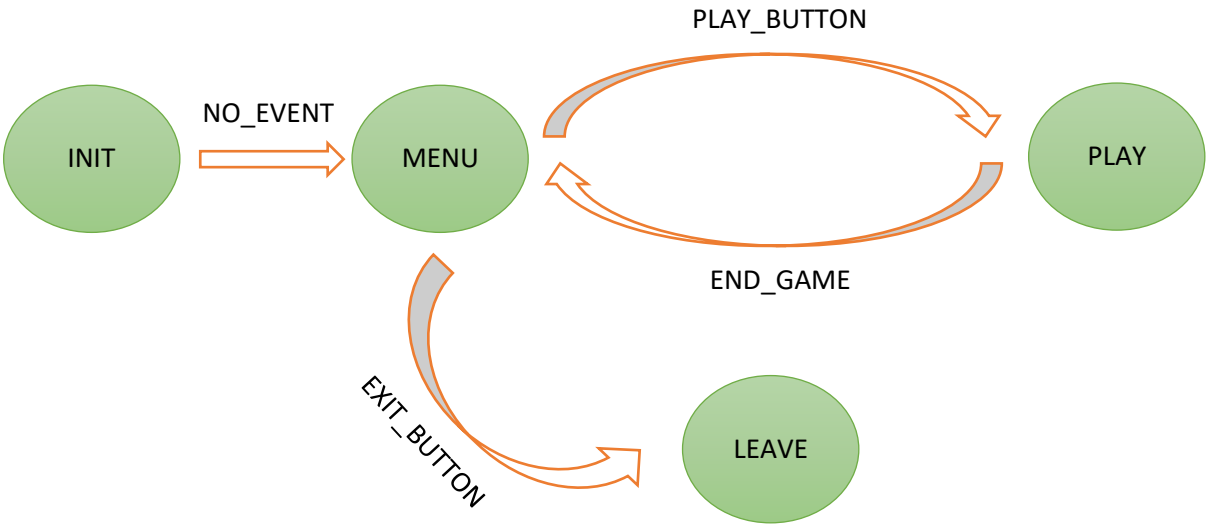
Dispositivo Entrada/Saída	Utilização	Uso de interrupções?
Timer	Atualização do jogo	Sim
Teclado	Controlo da cobra	Sim
Rato	Interação com menus e controlo do cursor	Sim
Placa Gráfica	Desenho de imagens	Não
RTC	Gravação do historial de jogos	Não

Organização / Estrutura do código

Game

"Game" é o módulo principal da aplicação, através do qual o jogo funciona, e onde praticamente toda a execução do programa ocorre. Possui funções primárias como as de inicialização do jogo e dos objetos que o compõem, nomeadamente as palavras, a cobra e o cursor.

Além disso é também neste módulo que se encontra implementada a máquina de estados que gere o programa e a função que os relaciona através de diversos eventos – *handle_event*. Em seguida apresenta-se uma figura ilustrativa desta mesma máquina de estados.



O jogo é iniciado no estado INIT, seguindo-se uma transição para o menu principal, MENU. As principais funções são então *main_menu* e *play_game*, sendo todas as outras invocadas através destas. No final de um jogo, retorna-se ao menu principal e pode sair-se, dirigindo-se o sistema para o estado LEAVE.

Módulo desenvolvido por Sandro Campos | Peso Relativo: 30 %

VBE

Módulo resultante do trabalho realizado nas aulas laboratoriais da placa gráfica, contém a função *vbe_get_mode_info* necessária na inicialização do modo de vídeo.

Peso Relativo: 3 %

Video_graphics

"Video_graphics" é um módulo importante na vertente gráfica do programa e intimamente associado aos anteriores. É através deste que conseguimos inicializar o modo gráfico do sistema operativo MINIX, além de desenhar imagens e pixéis isolados no ecrã que por fim nos permitem obter o cenário de jogo e alterar as cores dos diversos componentes. Este contém ainda as funções necessárias para o uso da técnica de *double buffering* referida anteriormente.

Módulo desenvolvido por ambos | Peso Relativo: 15 %

stb_image

Este módulo foi-nos útil permitindo-nos trabalhar com imagens no formato PNG adquirindo diversas vantagens, desde podermos usar um maior leque de cores à conveniência e facilidade com que se podem tratar este tipo de imagens em programas de edição de imagem. O código-fonte é da autoria de Sean Barrett e encontra-se publicado no seguinte endereço (*https://github.com/nothings/stb*).

Peso Relativo: 10 %

I8042

Módulo resultante do trabalho realizado nas aulas laboratoriais do teclado e do rato. Possui essencialmente diretivas #define necessárias à configuração específica de cada um destes periféricos.

Peso Relativo: 3 %

I8254

Módulo resultante do trabalho realizado nas aulas laboratoriais do timer. Possui diretivas #define inerentes ao funcionamento do timer e às suas caraterísticas.

Peso Relativo: 3 %

Timer

Módulo com código importado dos laboratórios do timer.

Peso Relativo: 8 %

Keyboard

Módulo com código de implementação do teclado. Contém nomeadamente um *handler* em linguagem *assembly* no ficheiro designado por *kbd_asm.S*.

Peso Relativo: 8 %

Mouse

Módulo com código importado dos laboratórios do rato.

Peso Relativo: 15 %

RTC

Neste módulo encontramos a implementação do periférico *Real Time Clock*. Faz-se uso de 2 registos, RTC_ADDR_REG e RTC_DATA_REG, que nos permitem obter informação acerca da hora e data a um determinado momento. Para este efeito invocam-se as funções *rtc_time* e *rtc_date*, ambas no interior da função *write_winner* em *game.c*.

Módulo desenvolvido por Bruno Fernandes | Peso Relativo: 5 %

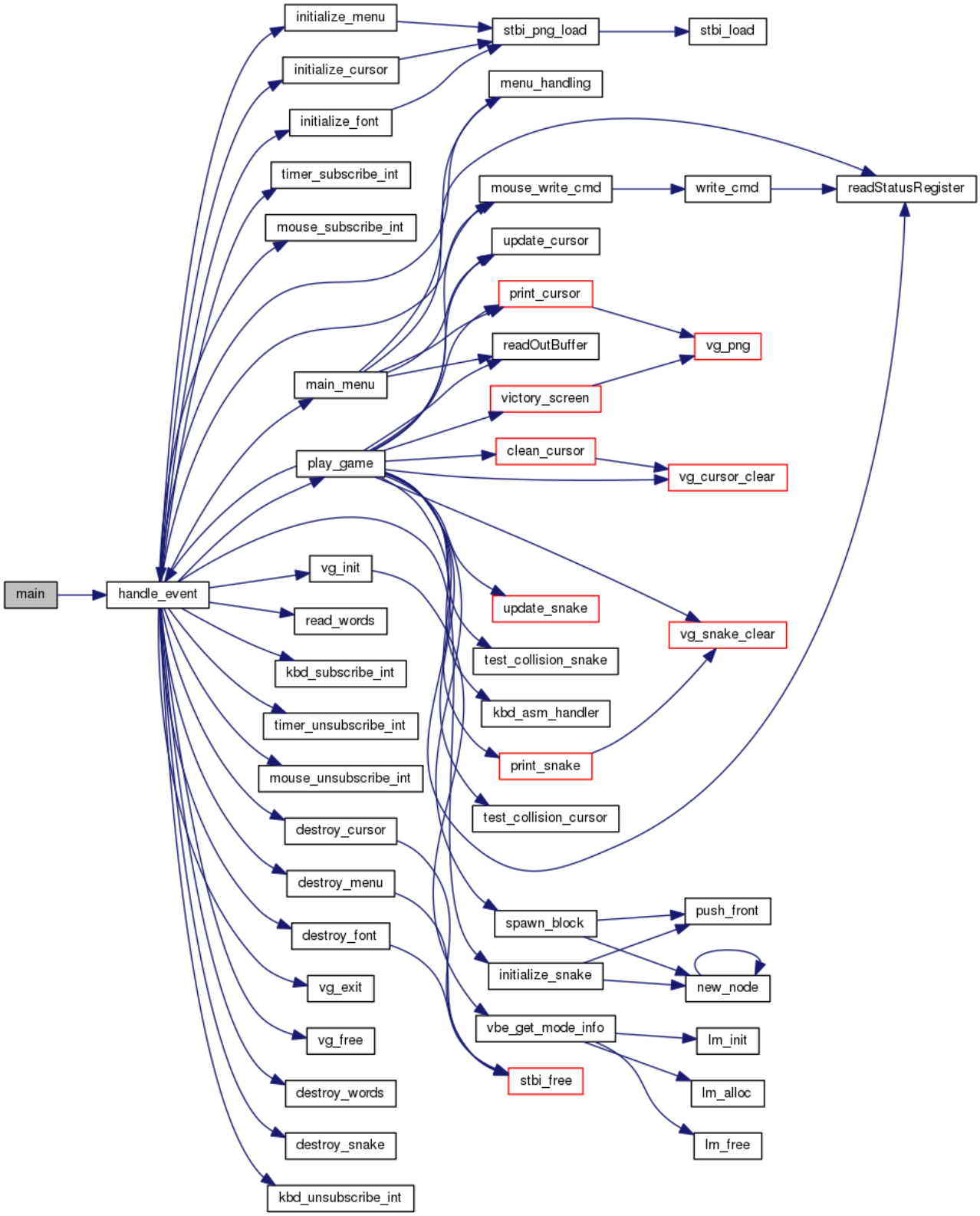
Detalhes da implementação

Um dos aspetos mais interessantes e desafiantes da nossa implementação prende-se à própria lógica de jogo e aos testes de colisões.

No que diz respeito à cobra, foi necessária a criação de um novo tipo de estrutura, a lista ligada. Esta, na nossa opinião, é aquela que mais se adequa ao tipo de operações que desejamos fazer – adicionar ou remover elementos do início ou do fim do *array* de blocos da cobra, enquanto cada bloco mantém uma ligação aos seus blocos vizinhos. Por sua vez, implementaram-se as funções *push_front*, *push_back*, *pop_front* e *pop_back*.

Outra implementação que exigiu alguma reflexão foi a de organizar as diversas letras geradas pelas palavras, e de que forma se poderiam gerar coordenadas aleatórias para cada uma, sabendo que há sempre restrições a cumprir. Houve necessidade então de se gerarem apenas coordenadas que não coincidissem com as de outras letras da mesma palavra, nem com o local onde se encontra presente a cobra. Decidiu-se, além disso, que o *array* escolhido para acolher as palavras seria partilhado entre a cobra e o cursor, contendo as letras da palavra no seu início e o resto do seu espaço servindo para guardar as letras aleatórias adicionais, geradas para o cursor.

Function call graph



Conclusões

Concluindo este projeto, ao longo de várias semanas de intenso trabalho, pensamos ser notória uma evolução significativa do nosso conhecimento em relação a diversas áreas envolvidas na produção de aplicações. A implementação de periféricos, desde os mais simples como é o caso timer, à interação gráfica com a placa de vídeo um pouco mais complexa, permitiu-nos elaborar aquela que é a nossa primeira aplicação interativa e com interface gráfica, o que muito nos entusiasmou. Neste sentido, gostaríamos de referir um aspeto particularmente positivo da unidade curricular, o incentivo à produção criativa dos alunos.

Além disso recorremos ao uso da linguagem C, que reconhecemos ser muito poderosa e flexível, mas por outro lado pensamos que tenha dificultado de alguma forma a construção de um código curto e conciso podendo ter consequentemente afetado a modularidade do mesmo.

Em síntese, gostaríamos de realçar que este projeto foi muito enriquecedor e deveras exigente, obrigando a um elevado empenho associado a muitas horas de trabalho árduo.

Apêndice

Instruções para instalação

Para dar início ao programa, através do ambiente MINIX, dirigir-se ao diretório "src" do projeto e executar ordenadamente os seguintes comandos:

1. **#sh install.sh**
2. **#sh compile.sh**
3. **#sh run.sh**

Com a execução do primeiro o programa é instalado no sistema, com o segundo é compilado e com o último finalmente executado.

Na instalação é efetuada uma cópia do diretório "res", que contém os recursos necessários à execução do programa (imagens e ficheiro de input "words.txt"), para uma localização conhecida e conveniente ("/home/snaktionary") - este diretório será também aquele em que será guardado o ficheiro "winners.txt".

Atenção, caso o ficheiro "words.txt" seja alterado por opção do utilizador, deve efetuar-se uma reinstalação do programa!

Gostaríamos ainda de referir que tivemos conhecimento deste processo de instalação através da consulta do blog de um ex-aluno do curso, Henrique Ferrolho (http://difusal.blogspot.pt/2014/07/minix-posts-index.html).