Laboratório de Computadores

LCOM – 2MIEIC07 – Grupo 4

Relatório do Projeto Final

LCOMFEUER

João Álvaro Cardoso Soares Ferreira – up201605592@fe.up.pt

Pedro Hugo Lima Noevo - up201604725@fe.up.pt

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Table of Contents

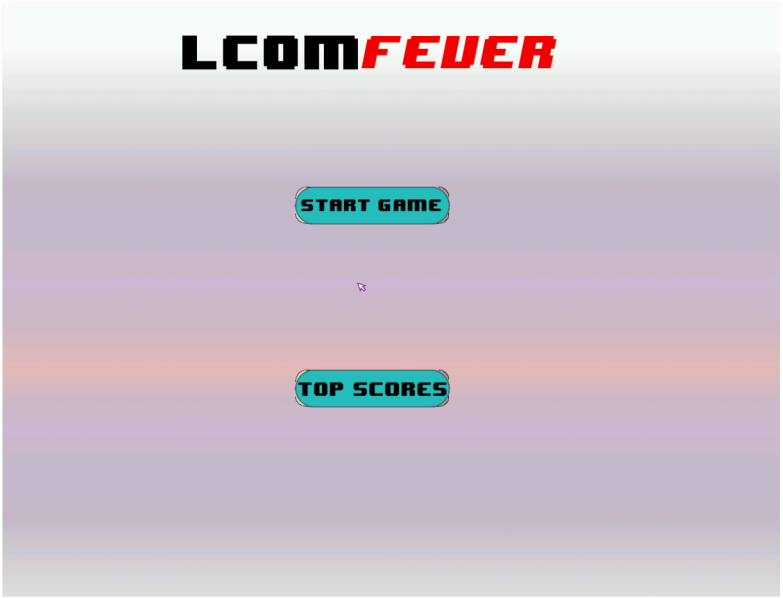
1 - Instruções de Utilização do Programa

1.1 Ecrã inicial

Ao iniciar o jogo, é apresentado ao utilizador um ecrã inicial para o receber. Este tem apenas o propósito de apresentar o jogo, sendo que o jogador pode fechá-lo imediatamente carregando na seta no canto superior esquerdo ou entrar na loja, o que o leva ao próximo menu.

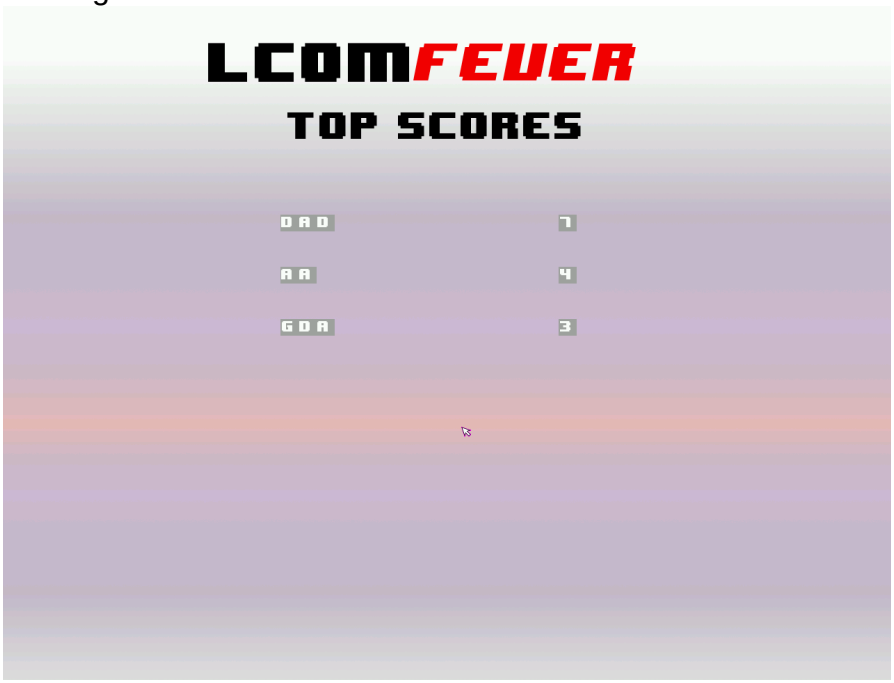
1.2 Menu inicial

O utilizador no menu inicial tem a opção de começar um jogo imediatamente, ver os registos das melhores pontuações já existentes (ambos utilizando o rato) ou sair para o ecrã inicial com o botão ESC do teclado.



1.3 Tabela de melhores pontuações

Na tabela de pontuações, tal como o nome indica, é possível ver o registo das melhores pontuações e dos nomes dos jogadores que as fizeram. Esta tabela altera-se após cada jogo, portanto o utilizador pode jogar uma ronda e vir a este menu imediatamente para ver como se compara aos restantes jogadores. Por motivos estilísticos, decidímos que o nome de cada jogador seria limitado a 3 letras. Para sair, o utilizador deve carregar no botão ESC.



1.4 Jogo

Quando o jogo começa, a loja abre e imediatamente surgem vários clientes, com velocidades diferentes, à procura de medicamentos. No canto superior esquerdo podemos ver o score atual e no direito o tempo, sendo que uma ronda do jogo dura 30 segundos. Temos em baixo o balcão com os comprimidos que serão então dados à clientela dependendo dos seus sintomas, que dirão quando pararem à frente do balcão.

Na imagem em cima podemos ver duas clientes que declaram os seus sintomas e esperam que o utilizador lhes dê os comprimidos apropriados. Podemos ver em cima que alguns comprimidos já desapareceram, tendo sido dados a outros/as clientes, mas o stock é reposto premindo a barra de espaço.

Cada cliente pode apenas receber dois comprimidos e um comprimido pode ter cura para um sintoma ou dois. Como tal, é necessário ter cuidado e ver quais os comprimidos corretos a dar a cada cliente, algo que podemos ver na tabela com o significado dos comprimidos, carregando na tecla A. Ao receber um comprimido cuja cura trate um dos seus sintomas, esse sintoma passa a “CURED”, sendo que quanto mais sintomas curados mais a pontuação recebida. Os clientes, os seus sintomas, a sua velocidade e as suas restantes propriadades são geradas aleatóriamente.

1.6 Pós-jogo – Guardar score

No fim dos 30 segundos, a ronda acaba e o utilizador depara-se com o seguinte ecrã:
Neste, o nome AAA é alterado com as teclas WASD. O utilizador começa na primeira letra e pode alterá-la com W e S (sendo que um passa para o caracter seguinte em ASCII e o outro para o anterior) ou alterar a letra em que está com A e D (A andando para a

esquerda e D para a direita). Terminando prime a barra de espaço, o que o retorna ao menu principal e adiciona o seu score à lista de high scores (se for alto o suficiente).

2 – Project Status

Tinhamos planeado a utilização de RTC e de Serial Port no trabalho final, interagindo com o jogo de modo a oferecer-lhe complecidade – o RTC serviria para aceder a valores de data e hora, o que seria util para o jogo, e o Serial Port para implementar um modo de dois jogadores. Devido a limitações de tempo e conflitos com outras unidades curriculares, tal não foi possível.

Apesar disso, todos os outros periféricos foram implementados com sucesso e funcionam como esperado. O projeto final não contém nenhuma

Device	Utilização	Interrupts?
Timer	Atualização de variáveis de jogo, chamada de algumas funções para a lógica de jogo e atualização dos gráficos	Sim
KBD	Navegação de menus, abrir/fechar tabela durante o jogo, input do nome no fim do jogo	Sim
Mouse	Navegação de menus, arrastar objetos durante o jogo	Sim
Video Card	Representação visual do jogo	Não

2.1 Timer

O timer é utilizado para atualizar os gráficos a cada 60 segundos e interagir com determinadas partes da lógica do jogo, nomeadamente a contagem decrescente para cada cliente e o seu movimento automático dependendo do estado do jogo. O timer também nos permite implementar animações ao longo do tempo como a seta que muda de cor no inicio do jogo ou o movimento dos jogadores.

2.2 KBD

Usado para a navegação de menus (carregando na tecla ESC retorna-se ao anterior), sair do jogo, abrir a imagem que mostra o efeito de cada comprimido durante o jogo, fazer reset à posição dos comprimidos durante o jogo e fazer input ao nome no fim.

O input do nome é feito com o teclado mas não é com input de texto direto, sendo através de substituição de caractéres com as teclas WASD. É dado ao jogador um nome pré-definido (AAA) que este pode modificar alterando cada caracter individualmente (a tecla W passa para a próxima letra enquanto que a S passa para a anterior, em ASCII) enquanto que as teclas A e D navegam entre os 3 caractéres.

Implementado no loop do jogo, que chama várias funções implementadas em keyboard.c e util.c

2.3 Mouse

Inputs do mouse são a principal forma de navegação de menus, sendo que é com estes que se carrega nos botões. Para além disto, o mouse é central ao jogo já que é com este que se arrastam os comprimidos para os clientes, sendo que para isto usamos tanto a posição do mouse como os botões premidos.

Implementado no loop do jogo, que chama várias funções implementadas em mouse.c e util.c

2.4 Video Card

A placa gráfica é essencial ao projeto, já que desenha todas as imagens que são representadas no ecrã (bitmaps no modo RGB (5:6:5)). O modo escolhido, de modo a ter o maior tamanho de um modo estável e económico, foi o modo 0x11A. Este modo tem dimensão 1280×1024 e uma palette de 16-bit . As imagens utilizadas retiradas de vários fontes na internet, sendo todas estas modificadas de modo a otimizar o seu uso por nós.

A font utilizada é 8 BIT WONDER.

De modo a que a representação do jogo seja sempre fluída, implementamos double buffer a cada 2 frames. Existe movimento de objetos mas sem colisão, já que a execução do jogo necessita que objetos se sobreponham – como tal, em vez de colisão, detetamos a sobreposição de objetos.

Implementado no loop do jogo, que chama várias funções implementadas em vbe.c, Bitmap.c, video_gr.c e util.c

3 – Organização e estruturação do código

3.1 Código desenvolvido por nós

highscores.c

Este módulo cria, desenha, lê e escreve de/para ficheiros, ordena e manipula de várias formas objetos Highscore que são usados para registar os pontos de jogadores.

Peso no trabalho: 10%

Membro responsável: Pedro Hugo

Estruturas associadas: Highscore (codificadas em Objetos.h por uma questão de organização)

timer3.c

Este módulo contem as funções relativas à subscrição de interrupts e obtenção/manipulação de informação do timer, desenvolvido durante a aula laboratorial sobre este.
Peso no trabalho: 5%
Membro responsável: João Álvaro e Pedro Hugo (50% cada)

keyboard_standalone.c
Este módulo contem as funções relativas à subscrição de interrupts e obtenção/manipulação de informação do keyboard, desenvolvido durante a aula laboratorial sobre este.
Peso no trabalho: 5%
Membro responsável: João Álvaro e Pedro Hugo (50% cada)

mouse.c
Este módulo contem as funções relativas à subscrição de interrupts e obtenção/manipulação de informação do mouse, desenvolvido durante a aula laboratorial sobre este.
Peso no trabalho: 5%
Membro responsável: João Álvaro e Pedro Hugo (50% cada)

video_gr.c
Este módulo contem as funções relativas à subscrição de interrupts e obtenção/manipulação de informação do mouse, desenvolvido durante a aula laboratorial sobre este.
Peso no trabalho: 5%
Membro responsável: João Álvaro

proj.c
Este ficheiro é responsável pelo início do jogo, criar e atribuir valor a todos as variáveis necessárias ao funcionamento do jogo (como Bitmaps que são carregados ou o ficheiro Highscore.txt que é interpretado). É também neste módulo onde está presente o loop do jogo com os respetivos interrupts para cada periférico e gestão do estado do jogo.
Peso no trabalho: 30%
Membro responsável: João Álvaro e Pedro Hugo (50% cada)

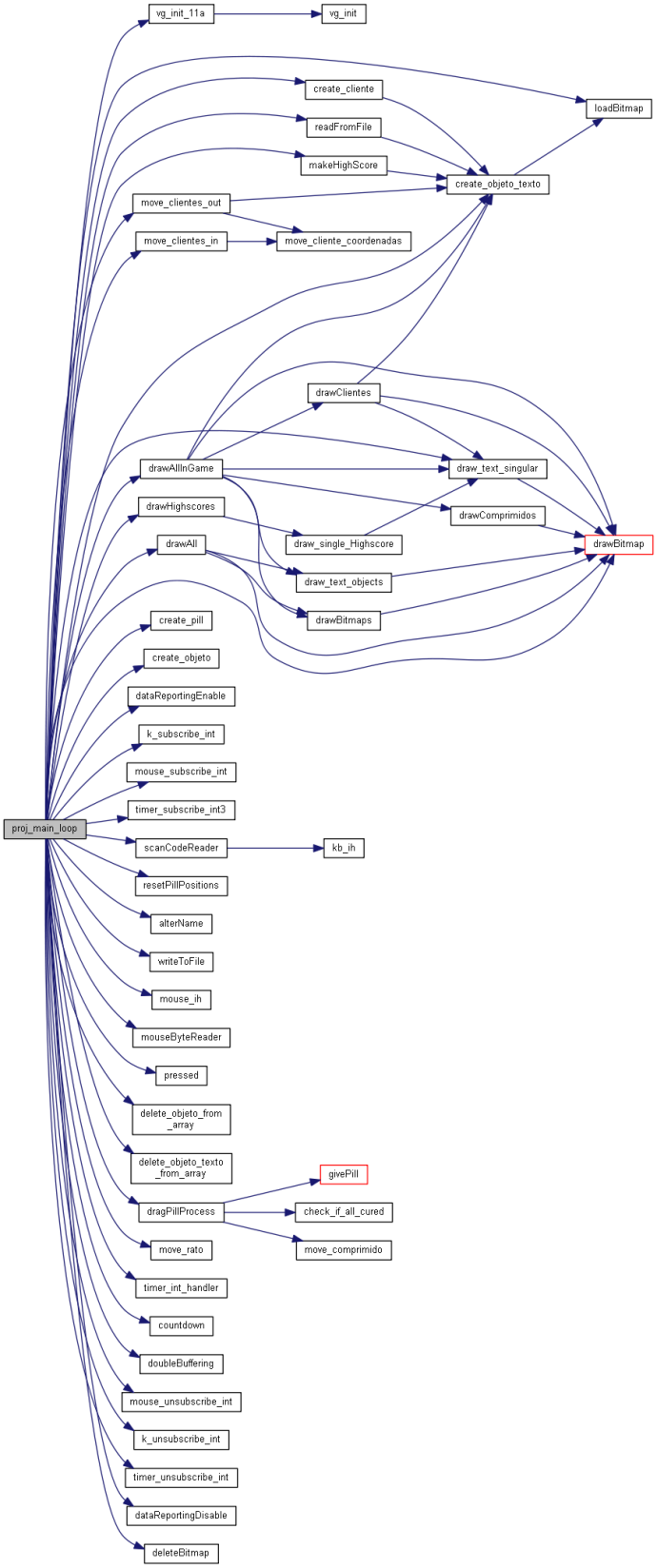
util.c
Neste módulo estão contidas quase todas as funções relevantes para a lógica do jogo, manipulação de dados utilizados pelo jogo, verificações, gestão dos estados, desenho de elementos num array, criação/edição/remoção de todo o tipo de objetos etc.
Peso no trabalho: 35%
Membro responsável: João Álvaro e Pedro Hugo (50% cada)
Estruturas associadas: Rato, Objeto, Objeto_texto, Cliente, Comprimido (codificadas em Objetos.h por uma questão de organização)

3.2 Código retirado da internet

bitmap.c
Este módulo é responsável por carregar para a memória do programa e desenhar Bitmaps que lhe são fornecidos. A função de desenho de Bitmaps foi alterada de modo a que estes possam ter transparência em zonas que estejam preenchidas pela cor 0xff00ff.
Peso no trabalho: 5%
Autor: Henrique Ferrolho **URL origem:** http://difusal.blogspot.com/2014/07/minix-posts-index.html

3.3 Call graph do loop de jogo

Devido ao seu tamanho impossibilitar uma visualização detalhada neste ficheiro, o gráfico será também guardado na pasta Docs.



4 – Detalhes da Implementação

4.1 Material coberto nas aulas

Dentro do material coberto nas aulas e disponibilizado pelos recursos da unidade curricular, utilizamos as seguintes técnicas:

- State machine – servimo-nos de state machines para distinguir as várias fases do jogo e o menu onde está o jogador, sendo que cada interrupt de um periférico diferente terá funções diferentes dependendo do estado atual do jogo.
- Double buffer – escrevemos todos os bitmaps para um buffer auxiliar que é apenas trocado para a VMEM de 2 em 2 frames, o que torna o movimento das imagens no projeto bastante fluído e sem falhas.
- Object orientation – servimo-nos de bastantes structs que são tratadas como objetos pelo nosso jogo, já que são extremamente uteis para a organização e manipulação de informação.

4.2 Material não coberto nas aulas

O overlapping de objectos, muitas vezes de diferentes tipos, foi um tópico que exploramos bastante neste trabalho através de recursos exteriores.

5 – Conclusões

Tendo chegado à conclusão do nosso projeto, podemos afirmar que este foi bem sucedido apesar de não termos conseguido implementar todos os periféricos que desejávamos (e, com eles, algumas funcionalidades no jogo). Estamos, portanto, satisfeitos com o resultado final, sabendo no entanto que com um pouco mais de tempo e menos conflitos com outras unidades curriculares poderia ser ainda melhor.

Desenvolvendo o projeto conseguímos chegar a um novo nível de compreensão sobre as interações entre os vários periféricos e sobre o desenvolvimento de projetos mais extensos em programação de baixo nível.

5.1 Avaliação da unidade curricular

Sendo ambos os membros do grupo do 3º ano, um de nós estando a repetir a unidade curricular e o outro tendo transferido de eletro, a experiência do nosso grupo em LCOM foi bastante única relativamente à maioria dos nossos colegas que a está a fazer pela primeira vez.

É evidente que a unidade curricular de LCOM é extremamente importante já que nos ensina sobre programação low level de forma prática e nos leva a interagir com periféricos e perceber como estes funcionam. O conhecimento e facilidade de manipular informação a um nível tão baixo é importate para variadíssimas áreas dentro da Informática. Tendo um dos membros do grupo feito a UC no ano anterior, podemos dizer que a introdução da LCF foi uma grande ajuda e melhoramente para a disciplina.

No entanto, LCOM continua a ser uma unidade curricular extremamente exigente e capaz de gastar muitas horas de um aluno sem que este faça progresso ou sequer aprenda, já que é bastante comum termos erros que não conseguímos explicar mesmo após extenso Debugging. A possibilidade não perceber imediatamente os conceitos e ficar atrasado pode trazer imensas repercursões já que todos os labs e o projeto final requerem que os labs anteriores tenha sido concluídos, exacerbando dificuldades sentidas.

Concluíndo, ter trabalhado e estudado esta unidade curricular a fundo com sucesso este ano levou-nos a perceber melhor a importância dos seus conteúdos, mas também a extensão da sua exigência.

6 – Apêndices

Para correr o projeto, basta extrair-lo para a pasta /home/lcom e correr os comandos “make” e “lcom_run proj” na pasta src. Se o professor desejar gerar e visualizar documentação Doxygen e/ou um novo Call Graph, pode também apenas fazer mover o ficheiro Doxyfile da pasta doc para a pasta project e executar o comando “doxygen Doxyfile” dentro da pasta src da pasta proj.