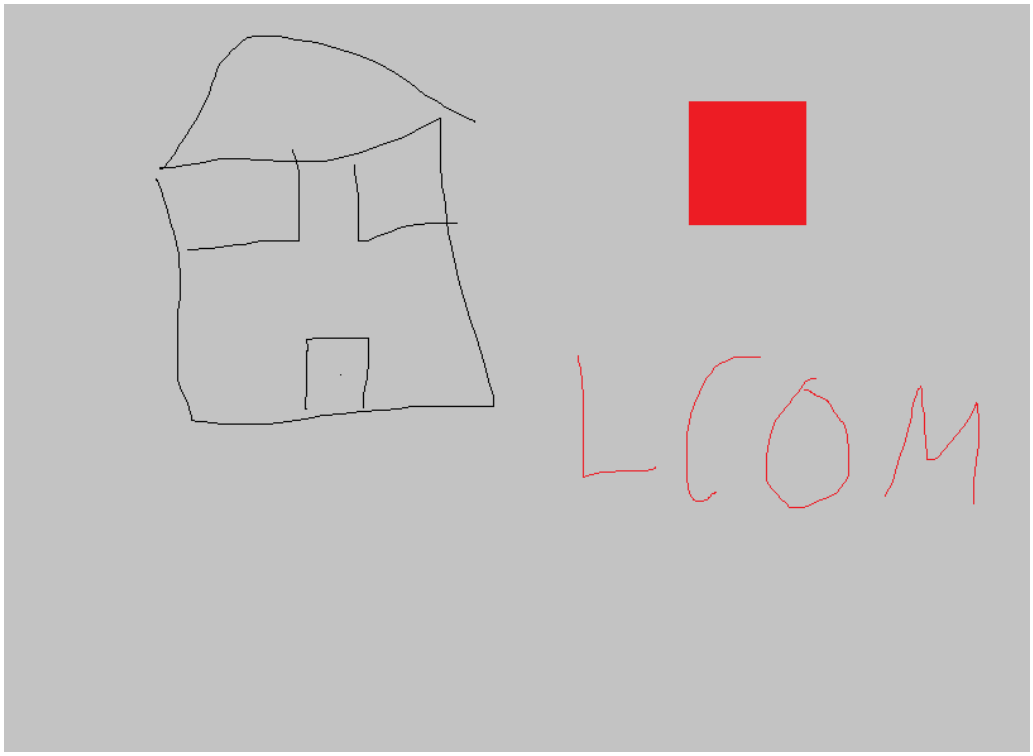


Artist Vs Guess

Por

Miguel Botelho up201304828
António Ramadas up201303568
Laboratório de Computadores

Índice

INSTRUÇÕES DO PROJETO

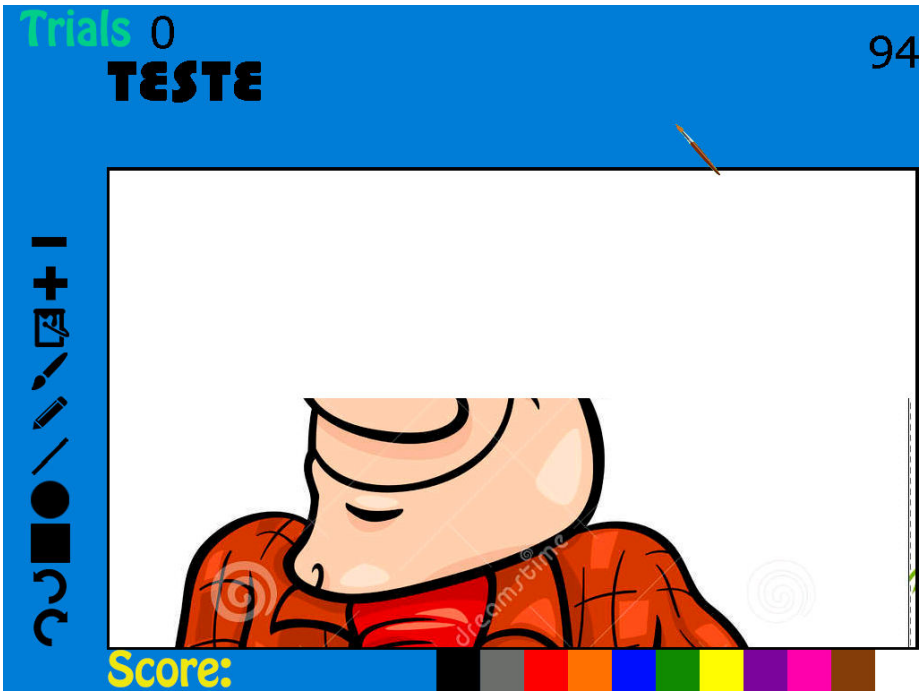
MENU INICIAL

No início do programa, é mostrado um menu principal que contêm 5 opções:

GET NAME

Pede o nome do utilizador, com o máximo de 10 letras. É possível apagar carregando no Backspace, apagando assim todo o username e sendo possível reescrevê-lo. Não são aceites números, símbolos, acentos ou espaços.

SINGLE PLAYER: HUMAN VS MACHINE



Inicia com o pedido do username do utilizador. De seguida, tem que adivinhar a imagem que aparece no ecrã, através do teclado com um máximo de 11 letras em que não há espaços. A imagem aparece de cima para baixo, desenhando cada vez mais da imagem e apagando o que foi desenhado, até que aparece toda a imagem. Durante este processo, é possível adivinhar já o desenho. Neste modo as ferramentas e as cores não estão ativas, sendo impossível desenhar na tela. Tem um número máximo de tentativas e, quando o utilizador achar que a palavra que escreveu é a resposta correta, basta carregar no Enter. Se for a escolha correta, é verificado se é um highscore. Se estiver errada ou o tempo máximo passar (99 segundos), leva-nos à perda do jogo. Carregando no ESC, volta ao menu principal.

MULTI PLAYER: ARTIST VS GUESS

Inicia com o pedido do username do utilizador. Agora é possível o jogador 1 fazer um desenho na tela enquanto o jogador 2 tenta adivinhar. Este desenho permite usar as seguintes ferramentas, por ordem:

- Menos: reduz o tamanho do que será desenhado
- Mais: aumenta o tamanho do que será desenhado
- Balde: pinta tudo da cor selecionada (apenas a cor onde o rato está)
- Pincel: desenha por pincel
- Lápis: desenha por lápis (linha)
- Linha: desenha uma linha carregando em 2 pontos com o rato
- Círculo: desenha um círculo no ecrã
- Quadrado: desenha um quadrado no ecrã
- Anular: volta atrás no desenho
- Refazer: se se anular, refaz o que foi apagado

E as cores:

- Preto
- Cinzento
- Vermelho
- Laranja
- Azul
- Verde

- Amarelo
- Roxo
- Rosa
- Branco

Se a palavra estiver correta (recorrendo ao teclado), basta carregar no Alt Tab, se estiver errada, carregar no Space. Novamente, tem um número máximo de tentativas e é verificado se é um highscore, com 99 segundos de tempo máximo. Carregando no ESC, volta ao menu principal.

ONLINE

Por muita pena nossa, não conseguimos implementar esta função do menu. Aqui seria utilizada a Serial Port e poderia jogar-se o jogo nos 2 computadores. Carregando no ESC, volta ao menu principal.

HIGHSCORES

Mostra os highscores que estão guardados num ficheiro do tipo .txt, que contém a data do highscore (usando o RTC), o nome e a pontuação, que é calculado pela fórmula: -135 * (segundos que passaram/ 60) + 10000 - número de tentativas utilizadas * 100. Carregando no ESC, volta ao menu principal.

EXIT

Fecha o programa.

LOST

O estado em que o programa fica se o utilizador passar o número de tentativas ou acabar o tempo. Carregando no ESC, volta ao menu principal.

WIN

A imagem que aparece quando se adivinha corretamente, juntamente com o nome do utilizador e a sua pontuação. Carregando no ESC, volta ao menu principal.

FUNCIONALIDADES DE CADA PERIFÉRICO

Periférico	Usado para	Interrupções
Timer	Refrescar ecrã, segundos para o Highscore	Sim
Keyboard	Escrever no ecrã	Sim
Mouse	Seleção dos menus e desenho	Sim
RTC	Tempo máximo de desenho	Sim
Video Card	Visualização de imagens, desenho e menus	Não

TIMER

O timer é usado para contar os segundos que passaram desde que o utilizador pode adivinhar a palavra e para refrescar o ecrã. É utilizado por interrupção e a sua implementação está no ficheiro timer.c e o seu uso no proj.c (linha 168 a 201).

KEYBOARD

O código do keyboard está no ficheiro keyboard.c e é usado no proj.c (linha 214 a 248) e um pouco no menu.c (para ver a key em que o utilizador premiu). A sua utilidade consiste no username do utilizador, adivinhar a imagem, e ir para o menu principal e funciona por interrupções.

RTC

O RTC é utilizado para contar a passagem do tempo máximo do jogo. Mais uma vez, é usado por interrupções e a sua implementação está em rtc.c e usa-se no proj.c (linha 334 a 357)..

VIDEO CARD

A Video Card é o único periférico que é utilizado por polling, refrescando o ecrã a cada tique do timer. A implementação está no ficheiro video_gr.c e é usado no menu.c (todo o ficheiro), frame.c (todo o ficheiro) e bitmap.c (linhas 3 a 138).

MOUSE

O Mouse é utilizado por interrupções e a sua função consiste em desenhar na tela e selecionar os menus. Está no mouse.c e é usado no proj.c (linhas 250 a 286), menu.c(apenas para ver a posição do rato atualmente) e mouse_struct.h.

SERIAL PORT

Tentamos implementar a Serial Port mas por polling não funcionava devido à sincronização de pacotes, isto é, como é necessário enviar bastantes pacotes num curto espaço de tempo, ocorriam imensas situações de erro, o que impossibilitou a utilização deste periférico por polling. No entanto, por interrupções, estas só eram geradas para o primeiro envia e mesmo quando limpávamos os buffers, estas não eram geradas. Encontra-se no ficheiro serial_port.c e é usado no proj.c (linhas 288 a 332).

ESTRUTURAÇÃO DO CÓDIGO

Antes de qualquer referência aos ficheiros, é importante salientar que cada ficheiro ficou responsável por ambos os elementos do grupo, não sendo feita qualquer tipo de divisão desse género.

ARRAY_KEYBOARD

Preenche os arrays que contém os bitmaps das letras e dos números. Este ficheiro foi feito 50% pelo António Ramadas e 50% pelo Miguel Botelho e é 1% do projeto.

BITMAP

Desenha bitmaps num buffer, faz load dos mesmos, delete e desenha com transparência (cor verde rgb(0,255,0)). Neste ficheiro foi utilizado código de Henrique Ferrolho. Assim, 35% foi feito por ele, 25% por António Ramadas e 5% por Miguel Botelho. Contêm 8% do projeto.

BMPFILE

Usamos este ficheiro para criar ficheiros do tipo .bmp. Este ficheiro foi encontrado na Internet e pode ser usado livremente. Foi alterado cerca de 10% dele por António Ramadas, pois continha alguns bugs e erros. 3% do projeto.

DEVICE INTERRUPTS

Todo este ficheiro foi feito por António Ramadas e trata das subscrições de todos os periféricos. 5% do projeto.

FRAME

Contêm as funções de desenho na tela, como desenhar quadrados, círculos, pincel e até mesmo redo (refazer) e undo (anular). 10% foi feito por António Ramadas e 90% por Miguel Botelho e faz parte de 8% do projeto.

GLOBAL VARIABLES

Contêm todas as variáveis globais. Foi feito por ambos os elementos e tem o peso de 1% no projeto.

KEYBOARD

Lê o input de uma tecla e transforma cada makecode no seu respetivo carater. 95% de António Ramadas, 5 % de Miguel Botelho e faz parte de 5% do projeto.

LIB

Este ficheiro apenas tem cada biblioteca que precisamos do Minix e de C.

MENU

Trata de todo o menu e respetivas opções. Verifica as palavras, bitmaps, a posição do rato relativamente ao menu, desenha os arrays do username e a palavra. Desenha ainda o tempo atual, tentativas e highscore. Seleciona as ferramentas de desenho e as cores. 5% foi feito por António Ramadas e 95% por Miguel Botelho e contêm 30% do projeto.

MOUSE

Preenche a struct do mouse, faz enable aos data packets e ao stream mode. 20% foi feito por António Ramadas e 80% por Miguel Botelho e faz parte de 5% do trabalho.

PROJ

É aqui se encontra o ciclo de atendimento de interrupções de todos os periféricos e os loads iniciais dos bitmaps. Lê também os packets do mouse e verifica se é o 1º, 2º ou 3º. Sai do programa, apaga a memória alocada e sai de modo gráfico. 30% foi feito por António Ramadas e 70% por Miguel Botelho e contêm 15% do trabalho.

READ_WRITE

Lê e escreve os highscores no ficheiro.txt. Este ficheiro foi feito na íntegra por António Ramadas e contêm 4% do projeto.

RTC

Dá enable e disable às interrupções do rtc, lê e escreve no mesmo. 90% foi feito por António Ramadas e 10% por Miguel Botelho.

SERIAL PORT

Reconhe interrupções e chama um interrupt handler quando é necessário enviar ou receber dados. O ficheiro foi feito, novamente, na íntegra por António Ramadas e faz parte de 0% do projeto.

STRUCT BMP

Este ficheiro contém a struct de Bitmaps de todas as imagens do menu.

STRUCT SCORES

Este ficheiro contém a struct dos 5 Highscores e de um Highscore.

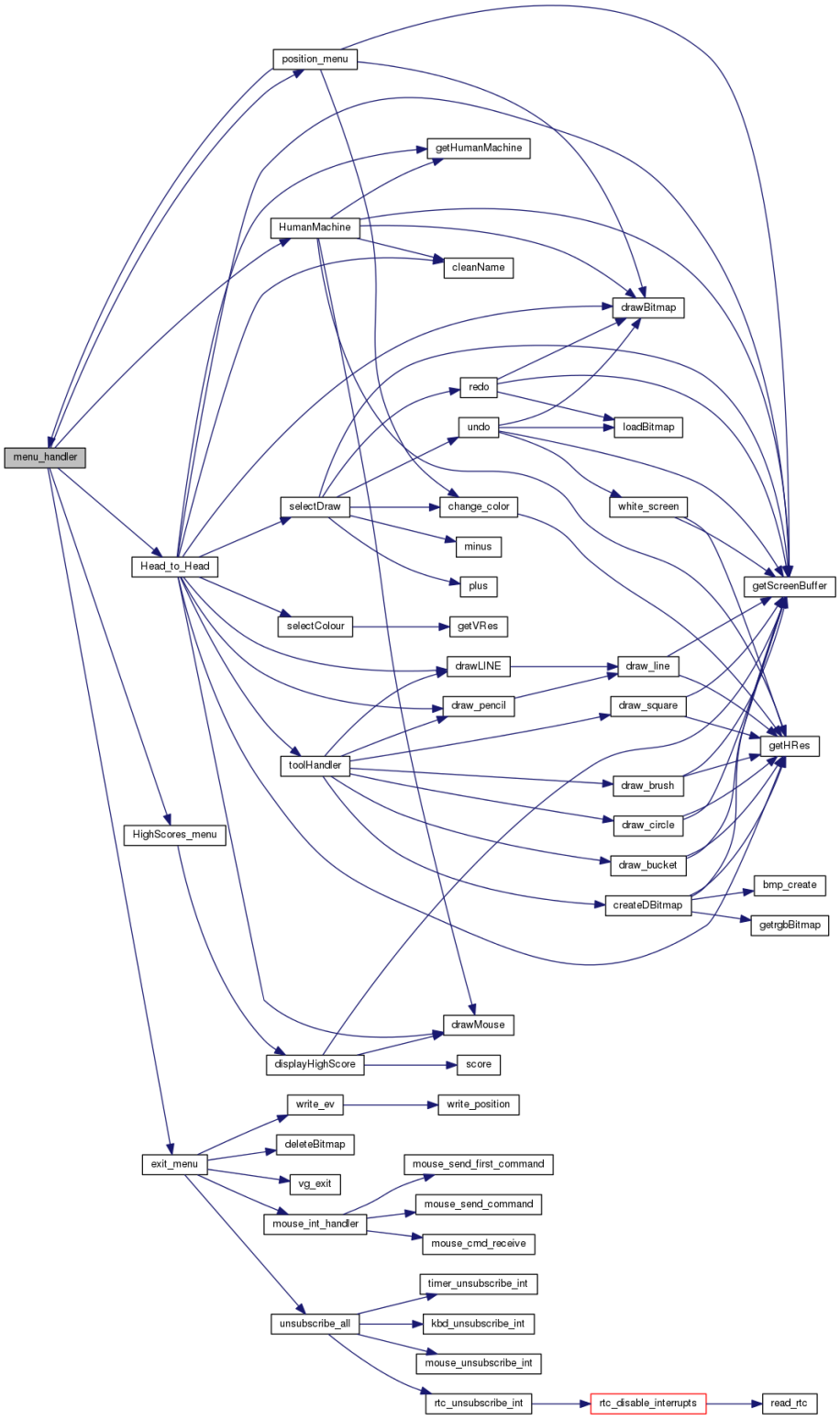
TIMER

Este ficheiro define a frequência do timer e obtém a sua configuração. Foi feito na íntegra por Miguel Botelho e faz parte de 2% do projeto.

VIDEO_GR

Inicializa o projeto em modo gráfico (modo 117 com 16 bits no nosso caso), obtêm a sua resolução e os bits por pixel, aloca a memória para todos os buffers, mapeia ainda a video memory, copia cada um dos buffers para o pretendido. Permite sair de modo gráfico e faz free aos buffers. Foi feito na íntegra por Miguel Botelho e contêm 10% do projeto.

GRÁFICO DAS CHAMADAS DE FUNÇÃO



IMPLEMENTAÇÃO

De um modo geral, quase tudo o que usamos no projeto foi coberto nas aulas de LCOM, como máquinas de estado, buffers, ciclos de atendimento, interrupções. No entanto, certos conceitos não foram bem interiorizados ou bem compreendidos. Damos por exemplo o conceito do "Double buffer" que foi dado na aula teórica mas era difícil compreendê-lo na altura, tendo sido lecionado muito cedo. Tivemos também alguns problemas ao nível das interrupções de certos periféricos. O mouse, por exemplo, funciona na perfeição no computador no António Ramadas mas, no do Miguel Botelho, é necessário reiniciar o Minix algumas vezes até ser possível dar enable aos data packets. Por outro lado, é preciso lembrar que não tivemos contacto suficiente com C durante o curso e, como tal, muitos erros surgiram por não termos muita interação com essa linguagem de programação e o que foi abordado, foi abordado numa das últimas aulas teóricas e já tínhamos necessidade disso para os labs. Tudo o que tivemos de aprender por nós próprios foi C e certos aspetos da Video Card, como mapear memória e semelhantes.

CONSIDERAÇÕES

A disciplina de LCOM é uma disciplina que ambos os elementos do grupo acham ser da maior importância para o curso, devido às suas aplicações práticas. Até agora, a única coisa que tínhamos feito era programação a alto nível (C++) ou a baixo nível (assembly), mas nunca tínhamos visto os resultados dessa mesma programação a este grau. Nesta cadeira conseguimos visualizar mais rapidamente os frutos do trabalho que tivemos. Na nossa opinião, é fulcral existir uma cadeira em que se lecione C antes de LCOM ou, pelo menos, aulas suplementares sobre a linguagem. É importante também referir que é possível, fazendo apenas o que nos é pedido em casa, acabar o resto do lab na aula. Contudo, isso é bastante difícil pois acontecem sempre erros que não estamos à espera ou conceitos com que não estávamos familiarizados. Assim, a preparação para os labs devia conter mais informação. E não menos importante, a disciplina vale poucos créditos de acordo com o que é preciso fazer para se perceber tudo o que se faz, dentro e fora da aula. Nós, como grupo, dedicamos sempre, aproximadamente, 10 horas ou mais por semana para a disciplina, algo que não corresponde aos créditos da disciplina.

Ambos os elementos consideram que o trabalho foi dividido por igual pelos dois.

É importante referir que é necessário colocar a biblioteca liblm.a (dada pelo professor no Lab 5: Video Card) e no ficheiro menu_macros.h mudar o define que muda o caminho dos ficheiros. Não usamos caminhos relativos e achámos que esta era a solução mais rápida e eficiente.

NOTAS

Um grande obrigado ao Henrique Ferrolho pelo tempo que disponibilizou ao ajudar muita gente este ano, ao facto de dar uma das aulas de LCOM e também de todas as suas sugestões relativamente ao modo gráfico, buffers e máquinas de estado. O código que usamos encontra-se neste site http://difusal.blogspot.pt/ na parte dos bitmaps, que foi a parte que usamos (modificamos a maior parte das funções dele pois descobrimos algumas incompatibilidades com o que queríamos fazer).

Os ficheiros bmpfile.h e bmpfile.c foram retirados daqui https://code.google.com/p/libbmp/source/browse/trunk/src/bmpfile.h?r=3 . Enquanto estávamos a fazer a pesquisa sobre bmps encontrámos este código público e livre que implementava (com alguns erros) o que prentendíamos.

REFERÊNCIAS

Para os ficheiros bmpfile.c e bmpfile.h:

https://code.google.com/p/libbmp/source/browse/trunk/src/bmpfile.h?r=3

Para os ficheiros bitmap.h e bitmap.c:

http://difusal.blogspot.pt/

Para o desenho de círculos:

http://en.wikipedia.org/wiki/Midpoint_circle_algorithm

Para o desenho de linhas:

http://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

Todas as fotos foram criadas a partir do programa Artist Vs Guess