U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

SUDOKU SAMURAI

turma 6 – grupo 9
2014/2015

INDICE

Pedro Fraga – up2012
Marta Lopes – ei12106

RESUMO

Desenvolvemos, no âmbito da unidade curricular de LCOM, o nosso projeto que se intitula de Sudoku Samurai e é uma versão do Sudoku original mas com um ambiente gráfico que tem como tema principal os Samurais.

O objetivo do jogo é preencher os quadrados vazios com números de 1 a 9 sendo que nenhum número se pode repetir em cada linha, coluna e quadrante. Existem 3 dificuldades associadas ao nosso jogo: *Easy, Medium* e *Hard,* sendo que as dificuldades variam com o número de quadrados já preenchidos inicialmente. Este jogo é jogado numa grelha de 9x9 quadrados divididos em 3x3 quadrantes.

Este projeto serviu para amplificar os nossos conhecimentos em volta de do *hardware* de periféricos de um computador. Sendo que os periféricos que usamos para o nosso projeto foram:

1. **Timer** para controlar as animações gráficas.
2. **Teclado** para o jogador escrever o número que pretende colocar no quadrado escolhido.
3. **Rato** para navegar pelos menus e para selecionar o quadrado que o jogador pretende preencher.
4. **Placa de vídeo** para mostrar o conteúdo gráfico do jogo (menus e grelha de jogo).
5. **RTC** para obtermos a data e apresentar no menu inicial.

A nível de *software* foi utilizado o seguinte:

1. **Minix** através do **VMware Player**.
2. **Eclipse**
3. **Redmine** da FEUP
4. **Adobe Photoshop CS6**

DESCRIÇÃO DO JOGO

Este jogo, Sudoku Samurai, é um jogo de concentração onde são aplicadas as regras de um simples jogo de Sudoku. Usamos o rato para escolher o quadrado que queremos preencher, e com o teclado escolhemos o número, e sem repetir o mesmo número na mesma linha, coluna ou quadrante temos que preencher a grelha inteira.

O jogo começa num menu inicial onde o jogador escolherá com o rato uma das seguintes opções:

1. *Play*
2. *Quit*

Nesse menu inicial também será possível ver os high scores (que dependem apenas do tempo que o jogador demora a concluir o jogo), a data e o tempo atual.

Se escolhermos a opção *Quit* ou pressionarmos a tecla ESC o programa termina.

Ao selecionarmos a opção *Play* entrará num novo menu onde podemos escolher a dificuldade do jogo (*Easy, Medium, Hard*) ou então ir para trás clicando em *Back*.

Quando é escolhida a dificuldade abrirá o ecrã de jogo onde terá a grelha com o jogo e à direita terá a contagem de tempo para depois calcular o high score do jogador. À medida que o jogador coloca os números nos quadrados pretendidos, irá haver avisos se o número colocado estiver errado (repetido na linha, coluna, ou quadrante). Após completar o quebra-cabeça irá surgir um novo ecrã com as palavras “You Win!” e com o botão *Back* para o jogador poder iniciar um novo jogo ou sair do jogo se desejar.

IMPLEMENTAÇÃO

ARQUITETURA DO PROGRAMA

Bitmap.c: Todas as funções relacionadas com bitmaps (carregar bitmaps, double buffering, etc...). Este código foi implementado pelo aluno do 3º ano Henrique Ferrolho, mas fizemos algumas adaptações, tais como desenhar um bitmap no double buffer em vez de desenhar na memória virtual.

bitmap.h

i8042.h

i8254.h

Jogo.c: [MOTOR DE JOGO] Neste ficheiro estão todos os menus de jogo, e funções relativas à funcionalidade de jogo, o programa está na maior parte das vezes a correr funções implementadas neste ficheiro. Fazem-se as verificações relacionadas com as regras do jogo e recebem-se notificações dos periféricos em cada um dos menus. Carregamos toda a informação dos ficheiros ".txt" respetivos para ser possível apresentar os high scores, e atualizamos essa informação cada vez que um jogo acaba.

kb.c : Ficheiro onde se encontram todas as funções que fazem o teclado funcionar (subscribe e unsubscribe).

kb.h

mouse.c: Ficheiro onde se encontram todas as funções que fazem o rato funcionar (subscribe, unsubscribe, ativar e desativar).

mouse.h

proj.c: Neste ficheiro inicializam-se todas as estruturas de dados, estas fazem parte de uma estrutura de dados "game". Definem-se também todas as coordenadas (pixel x, pixel y) de todos os quadrados no tabuleiro de sudoku, de outra maneira era impossível saber onde colocar o número em cada quadrado.

proj.h

rtc.c: Ficheiro onde se encontram todas as funções que fazem o RTC funcionar (subscribe, unsubscribe, set).

rtc.h

sudoku.h: Definição da nossa estrutura de dados "game".

timer.c: Ficheiro onde se encontram todas as funções que fazem o timer funcionar (subscribe, unsubscribe).

timer.h

vbe.c: Chamamos a função 01 do VBE, e reservamos memória para poder usar o modo especificado, temos também uma estrutura com todas as informações sobre o modo utilizado.

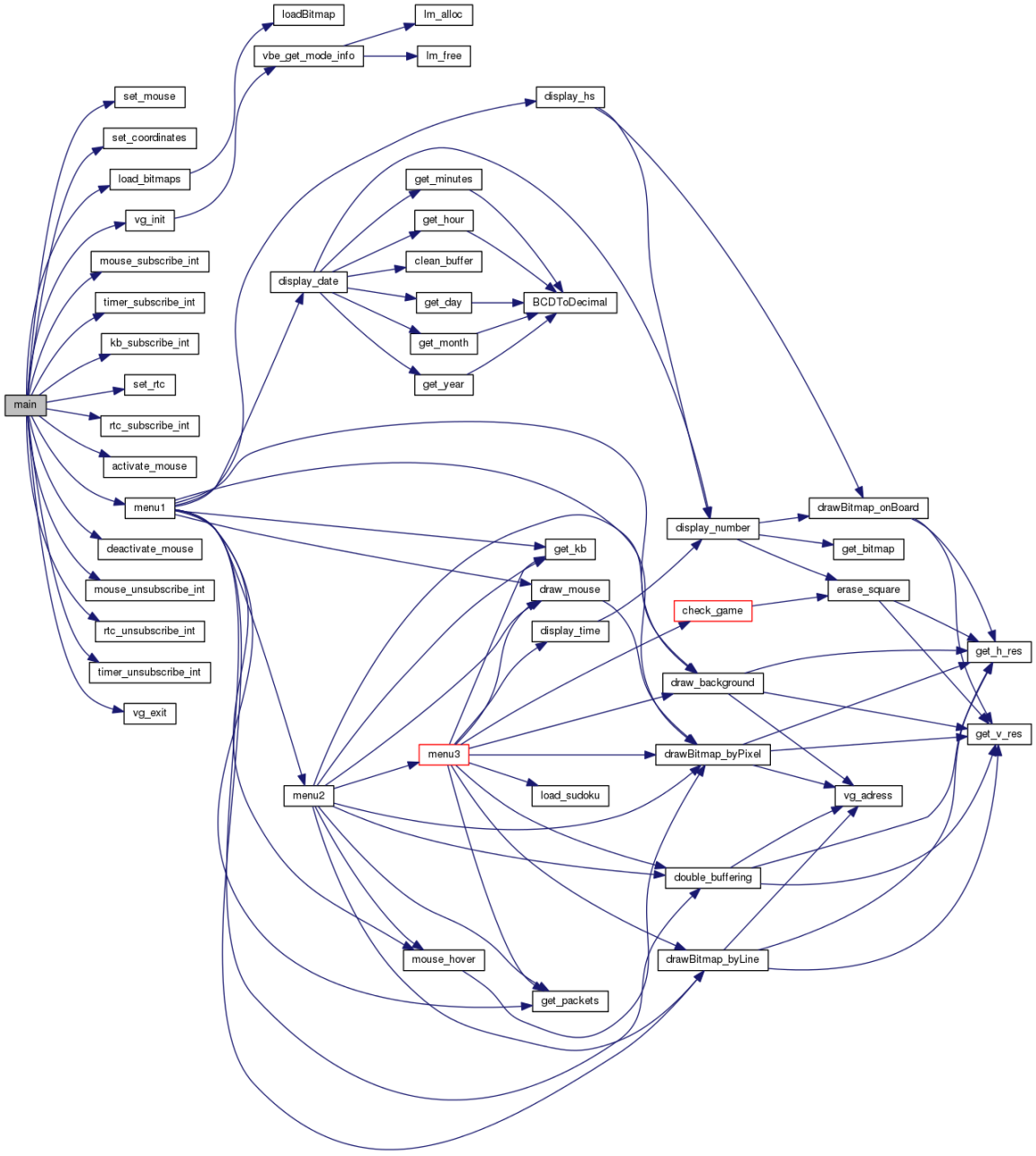
vbe.h

video_card.h

video_gr.c: Ficheiro onde se encontram todas as funções que possibilitam entrar e sair do modo gráfico.

vídeo_gr.h

GRAFICO DE FUNÇÕES



OUTRAS IMPLEMENTAÇÕES

Para uma execução mais fluída implementamos a técnica de double buffering, que guarda a memória virtual não dinâmica (que não vai ser alterada) em cada menu, não sendo preciso depois voltar a desenhar todos os bitmaps que não vão ser alterados.

ESTRUTURA DE DADOS

Implementamos uma struct “game”, que guarda todos os irqs, dados sobre o mouse (coordenadas no ecrã e botões pressionados), tem uma struct de bitmaps (local para onde são carregados todos os bitmaps para a execução do programa), e por fim contém também uma struct “sudoku”, com um array tridimensional “[coluna][fila][quadrado]” que guarda todos os números existentes no tabuleiro (0 por defeito) e o tempo gasto naquele jogo.

INSTRUÇOES DE EXECUÇAO

Podemos executar o programa com a ajuda dos scripts implementados, install.sh, compile.sh e clean.sh. Usamos o comando “sh install.sh” para copiar o projecto para um diretório temporário e copiar todos os ficheiros indispensáveis para a execução do programa. Usamos o “sh compile.sh” para compilar o programa e executar o programa. Depois usamos o “sh clean.sh” para apagar o diretório temporário.

AVALIAÇÃO DA UNIDADE CURRICULAR

No geral achamos que a unidade curricular está bem organizada, ajudou-nos claramente a perceber alguns conceitos relacionados com programação que antes não estavam esclarecidos, apenas julgamos que talvez a data de entrega de cada LAB deveria ser igual para todos e não dependendo do dia em que o grupo tem a aula, porque isso leva a que as pessoas que têm aula no final da semana tenham mais tempo para completar o LAB.

RESPONSABILIDADE POR MÓDULO

Marta Lopes

vbe.c
video_gr.c
kb.c
mouse.c

Pedro Fraga

timer.c
rtc.c
jogo.c
proj.c
Bitmap.c

CONCLUSÃO

No desenvolvimento deste jogo as nossas maiores dificuldades foram o rato e carregar os bitmaps para o jogo, no entanto a lógica de jogo por si mesma não trouxe grandes dificuldades visto que o Sudoku é um jogo com uma base simples. Como o nosso jogo não tem grandes movimentos ou animações dedicamos algum tempo à volta da parte gráfica para que fosse atrativa e interessante. Podemos dizer que estamos bastante satisfeitos com o nosso projeto e que gostamos de o desenvolver.