



Laboratório de Computadores

José Marques; Pedro Vieira de Castro; Vânia Leite
Mestrado Integrado em Engenharia Informática
3 de Janeiro de 2014

Introdução

Planeava-se que o projeto do grupo fosse o Minesweeper, com 2 modos de jogo (singular e multiplayer, recorrendo ao uso da porta de série). Todos os periféricos que se planeava usar foram integrados com sucesso. Contudo, havia 2 extras do jogo que não foram implementados: Não foram usadas gravações de histórico em documentos txt e não foi introduzido o item da bomba, por falta de tempo. Planeava-se que o teclado fosse usado para inserção do nome do jogador, mas tendo em conta que acabaram por não ser usados documentos txt, esta funcionalidade era um tanto inútil, por isso o teclado foi utilizado apenas para navegação de menus e reiniciar o jogo ou sair do mesmo.

Como balanço final, é apresentado um jogo do Minesweeper em que:

- A escolha da casa onde jogar é feita com recurso ao rato;
- São mostradas as horas ao utilizador no menu;
- Há modo singular ou multiplayer de jogo;
- Todos os desenhos são feitos com recurso ao `vb_draw_line`;

Instruções de execução

Para jogar, o utilizador deve escolher no menu o modo desejado:

- Singular

É gerado um mapa, e é feito o seu display a seguir. O jogador deve tentar adivinhar onde estão as bombas, clicando em casas em que ache que não haja uma bomba, para saber quantas bombas existem à volta dessa casa. Quando o jogador tiver a certeza que uma casa tenha uma bomba, clica sobre ela com o botão direito do rato de modo a plantar la uma bandada. Ganha quanse adivinhar onde estão todas as bandeiras. A qualquer momento do jogo, o jogador pode premir a tecla 'r' para reiniciar o jogo ou 'q' para desistir.

- Multiplayer
Após gerado o mapa, cada jogador tem a sua vez de jogar no mesmo mapa, ganhando aquele acertar mais bandeiras. Passa a vez de jogar quando um jogador erra ao colocar uma bandeira.

Arquitectura do programa

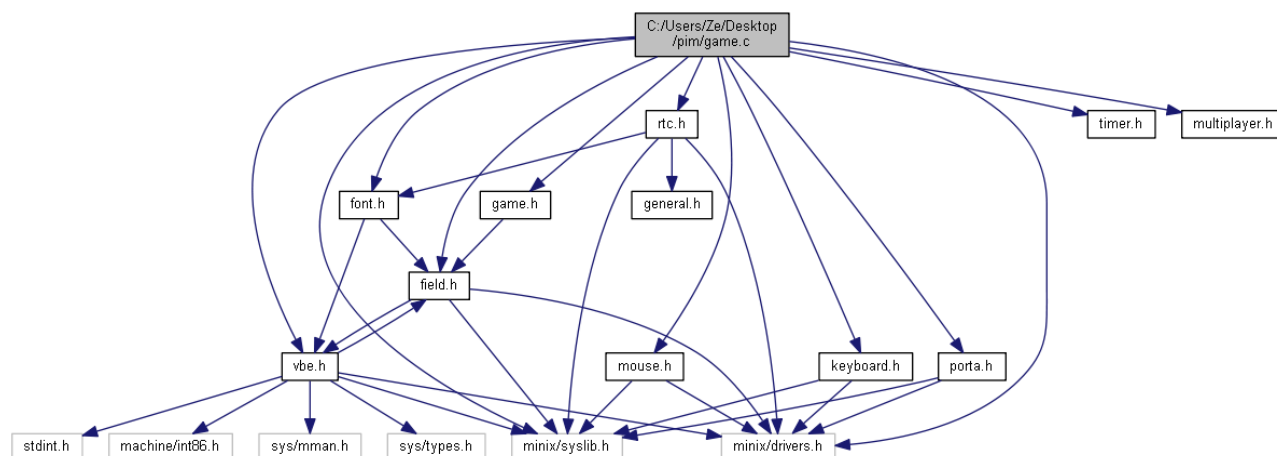
Módulos:

field.c	→	Usado para desenhos de figuras no ecrã
field.h		
game.c	→	Motor de jogo
game.h		
general.h		
keyboard.c	→	Teclado
keyboard.h		
liblm.a		
lmlib.h		
mouse.c	→	Mouse
mouse.h		
multiplayer.c	→	Motor de jogo do mode multiplayer
multiplayer.h		
porta.c	→	Porta de série
porta.h		
timer.c		
timer.h		
vbe.c	→	Modo gráfico
vbe.h		
fonts.c	→	Desenho das letras
fonts.h		
rtc.c	→	Real time-clock
rtc.h		

Estruturas de dados:

Foi usada uma struct para representar cada casa do tabuleiro, que contém a informação necessária para saber se está revelada, se tem uma mina, bandeira e o seu numero (numero de bombas à volta dessa casa ou -1 se for uma mina).

Diagrama de invocação de funções



Aspectos relevantes de implementação

Foram utilizadas interrupções no rato, time e keyboard, enquanto o rtc e porta de série não, porque o polling facilita a receção dos dados e quantidade necessária de acessos a estes periféricos não justifica o uso de interrupções.

É utilizado um buffer para controlar a imagem do ecrã, que é atualizado, com o auxílio do timer, 60 vezes por segundo.

Os mapas são gerados aleatoriamente a cada vez que se joga.

Apesar de se ter feito um programa capaz de lidar com diferentes lados de tabuleiro e número de minas, tal como a sensibilidade do rato, o tempo não permitiu implementar estas opções para ficarem disponíveis para o utilizador.

Auto-avaliação de cada elemento

Auto-avaliação José Marques

Código a cargo: timer e vbe

Participação: 1/4

Contribuição: 1/4

Auto-avaliação Pedro Castro

Código a cargo: porta, mouse, game, multiplayer

Participação: 1/2

Contribuição: 1/2

Auto-avaliação Vânia Leite

Código a cargo: Fonts, keyboard, field, rtc

Participação: 1/4

Contribuição: 1/4

