

Mestrado Integrado em Engenharia Informática e Computação
EIC0020-LABORATÓRIO DE COMPUTADORES

MOUSE MAZE



TURMA 3 - GRUPO 7

RUI VILARES - UP201207046

VITOR ESTEVES - UP201303104

2 JANEIRO DE 2015

Índice

Introdução	3
Explicação dos modos de jogo	3
Periféricos utilizados	3
Software/Tutoriais utilizados	3
Descrição do Programa	4
Implementação	4
Descrição da arquitetura do programa	4
Funcionalidades implementadas	5
Funcionalidades especificadas, mas não implementadas	5
Limitações Conhecidas do Programa	5
Mecanismos Relevantes da Implementação	5
Descrição do mecanismo de detecção de colisões	5
Descrição do mecanismo do modo night	5
Instruções de Execução	6
Call Graph	7
Conclusões	8

Introdução

O projecto que desenvolvemos ao longo do semestre, denominado **Mouse Maze**, é uma versão baseada no jogo original com o mesmo nome, atualmente disponível, sob a forma de aplicação, para varias plataformas incluindo IOS e Android.

O objectivo do jogo é ser capaz de chegar ao fim do labirinto sem tocar nas paredes do mesmo. Existem quatro modos de jogo: “Normal with Mouse”, “Normal with Keyboard”, “Night with Mouse” e “Night with Keyboard”.

Explicação dos modos de jogo

No modos de jogo “Normal” é possível jogar a versão clássica do Mouse Maze na qual controlamos o objecto com o teclado ou rato (opções “with keyboard” ou “with mouse”) de modo a chegar ao fim do labirinto.

No modo de jogo “Night” apenas é possível ver uma pequena porção do labirinto, obrigando-nos a “adivinhar” o caminho, tornando assim o jogo mais complexo.

O projecto foi desenvolvido com o objectivo de melhorar o nosso conhecimento a nível de hardware de periféricos de um computador, obrigando-nos a programar e manipular os mesmos desde o zero.

Periféricos utilizados

Os periféricos que utilizamos no nosso projecto são:

Placa gráfica, em modo de vídeo, para a interface gráfica do nosso jogo;

Keyboard, através de interrupções, para jogar nos modos “with keyboard” e em interações nos menus;

Timer para controlar o processamento e ditar o fluxo de eventos;

RTC para cronometrar o tempo do jogo;

Mouse para navegar nos menus e mover o objecto ao longo do labirinto nos modos “with mouse”;

Software/Tutoriais utilizados

Sistema Operativo Minix (3.1.8), através de uma máquina virtual criada com o VMware Player;

Eclipse Luna para programação;

Redmine (SVN) da FEUP para upload e gestão das diversas versões de código do nosso projecto;

Tutoriais encontrados em <http://difusal.blogspot.pt/>, escritos pelo monitor Henrique Ferrolho

Doxywizard para gerar a documentação Doxygen;

Adobe Photoshop CS6 na criação de botões e fundos;

Descrição do Programa

O **Mouse Maze**, é um simples jogo de labirintos onde o objectivo, é utilizar o teclado ou o rato para controlar o objecto de modo a ser capaz de atingir o final do labirinto, sem tocar nas suas paredes o mais rapidamente possível.

Antes de começarmos a jogar, é apresentado um menu onde o jogador poderá escolher, com o rato ou com as setas do teclado, uma das opções:

1. Play Game
2. Exit

Ao seleccionarmos a opção 1 entramos num novo menu onde podemos escolher entre os varios modo de jogo: “Normal with Mouse”, “Normal with Keyboard”, “Night with Mouse” e “Night with Keyboard”.
Se escolhermos a opção 2, o programa termina.

Implementação

Descrição da arquitetura do programa

main: Inicialização do modo gráfico, inicialização do labirinto e ainda ciclo de controlo do programa.

video_gr: Contém as funções de uma biblioteca gráfica desenvolvida por nós, tais como desenhar formas geométricas.

mouse-maze: Controlo da maquina de estados.

timer: Contém as funções de tratamento e utilização do timer.

keyboard: Subscribe e unsubscribe das interrupções do teclado.

kbc: Contém as funções que escrevem e lêem o estado do kbc.

rtc: Funções necessárias para a utilização do RTC.

mouse: Contém as funções de tratamento e utilização do rato.

labirynth: Funções de manipulação/criação de labirintos.

MainMenuState: Menu principal onde se pode escolher outro menu para onde navegar.

ModeState: Funções para manipular os modos de jogo.

LoseState: Funções de manipulação dos estados de derrota.

WinState: Funções de manipulação do estado de vitoria.

bitmap: Classe para carregar, destruir e visualizar bitmaps.

rectangle: Manipulação de formas geométricas.

Funcionalidades implementadas

Em relação à proposta inicial do projecto as funcionalidades implementadas foram:

Na versão alpha, existia um labirinto inicial onde foi implementada a lógica do jogo, timer, teclado e placa gráfica.

Na versão beta, foi já possível observar Menus e gráficos melhorados (através do uso de BMPs).

Foi possível observar as interrupções a partir do rato;

Foi possível observar as colisões entre os objectos, incluindo nos Menus Interactivos.

Existem 4 Modos de jogo, na qual é possível utilizar tanto teclado como rato para poder jogar.

Funcionalidades especificadas, mas não implementadas

Tal como referido na especificação do projecto, considerámos a porta de série opcional, acabando mesmo por não a implementar devido à sua complexidade acrescida.

Não foi também implementado o registo de highscores.

Limitações Conhecidas do Programa

Falta de níveis.

Mecanismos Relevantes da Implementação

Descrição do mecanismo de detecção de colisões

O mecanismo de colisões foi um dos mais complexos e importantes mecanismos utilizado no nosso trabalho. Através da implementação da classe rectangle (permite-nos manipular rectângulos e é o cerne do mecanismo) foi-nos possível encontrar a interacção necessária entre labirinto e objecto de jogo. Assim sendo, a nossa condição de colisão será sempre verdadeira enquanto o objecto do jogo colidir com um dos vários rectângulos que compõe o labirinto. Quando isto não se verificar significa que o jogador ultrapassou a margem, invalidando assim o progresso do jogo.

Descrição do mecanismo do modo night

O mecanismo utilizado no modo “night” consiste no preenchimento do ecrã por completo, excepto o círculo que permite a visualização do raio à volta do rectângulo de jogo.

Instruções de Execução

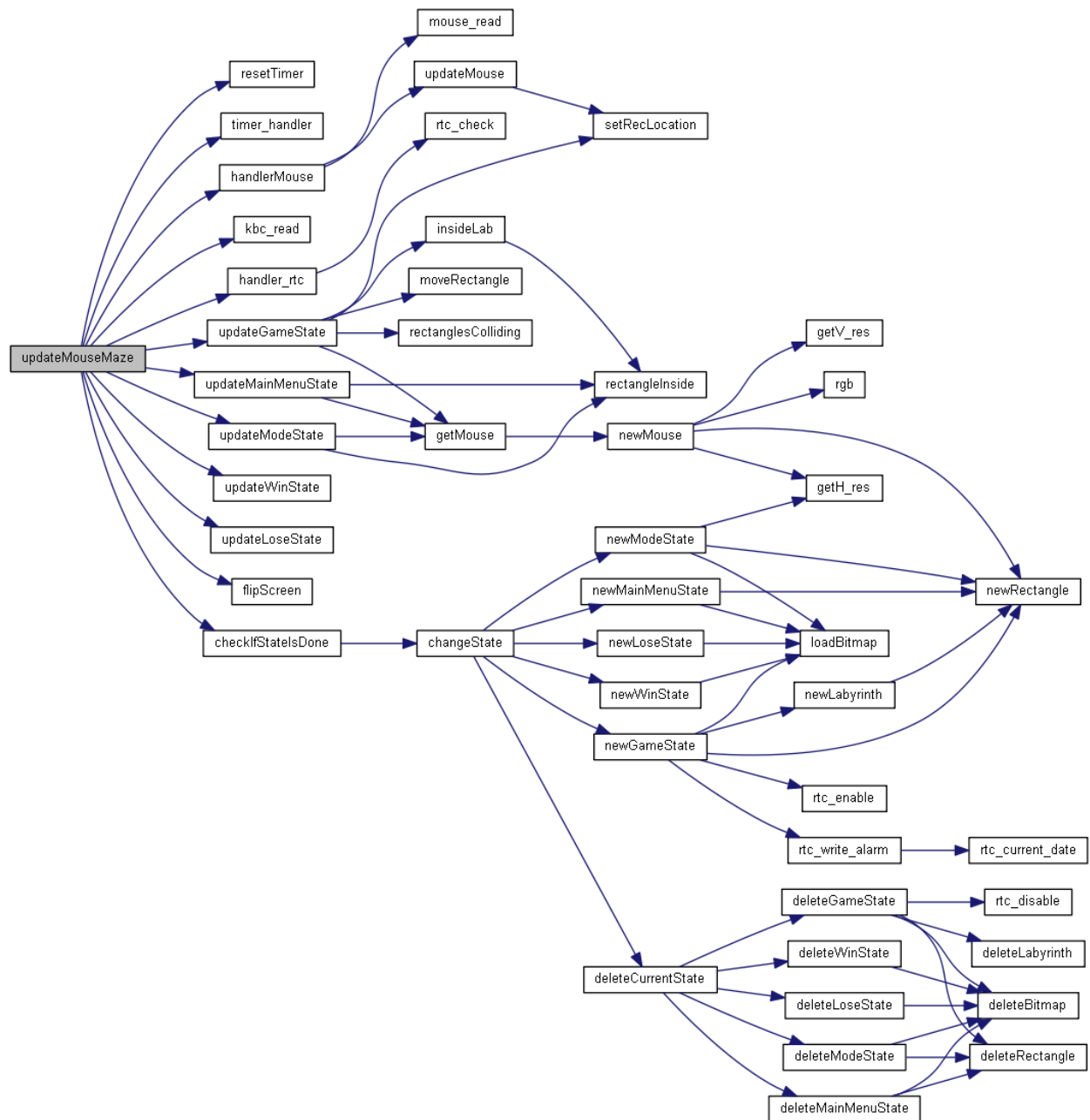
Antes da primeira execução é necessário copiar o ficheiro presente na pasta /conf para a pasta etc/system.conf.d/.

Para executar o programa, inicialmente, o utilizador deve alterar o modo para poder usufruir de privilégios de administrador executando o comando su.

Em seguida, após lhe ter sido garantido acesso root deve utilizar o comando:

```
service run /home/lcom/lcom1415-t3g07/proj/src/mouse-maze
```

Call Graph



Conclusões

Terminado o projecto, o saldo é positivo, pois foi-nos possível compreender melhor e utilizar de maneira mais eficaz o hardware dos periféricos.

Durante o desenvolvimento do projecto tentámos ao máximo focar-nos na manipulação do maior número possível de periféricos de modo a aprofundar o conhecimento adquirido ao longo do semestre, não nos preocupando muito com a adição de extras não relacionados directamente com os mesmos.

Não sentimos assim grandes dificuldades ao longo do projecto, estando bastante satisfeitos com o produto final do nosso trabalho.