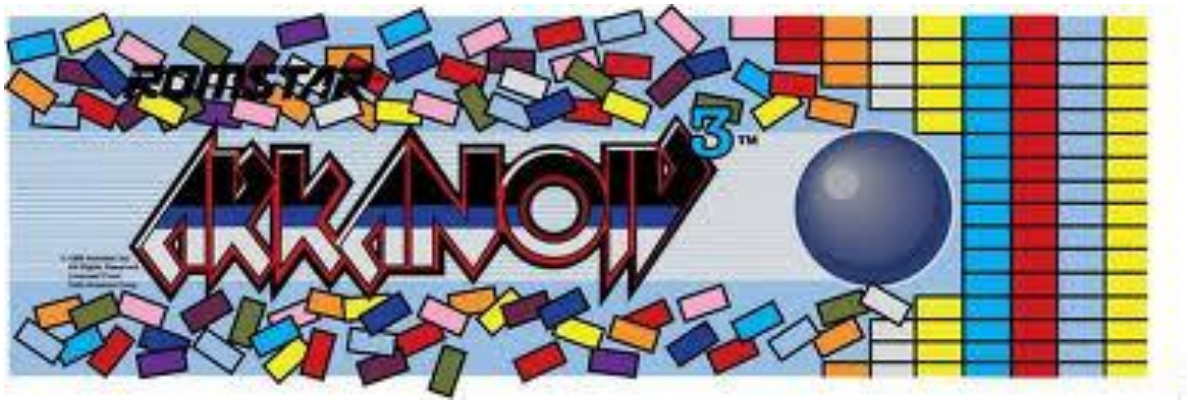
FEUP

EIC0020 - Laboratório de Computadores

2009/2010 - 2S



01-01-2011

Turma 5, Grupo 4

Autores:
Carlos Miguel Correia da Costa, ei09097, ei09097@fe.up.pt
Wilson Beto Amaral Pimentel, ei09052, ei09052@fe.up.pt

1. Resumo

No âmbito da disciplina de Laboratório de Computadores, foi proposto um projecto com tema escolhido por nós que tem como objectivo dar a conhecer e utilizar a interface de hardware dos periféricos de um computador, bem como melhorar a capacidade de trabalhar com software de baixo nível e adquirir prática em programação tanto na linguagem *C* como em *Assembly*.

A nível de hardware, foi utilizada a placa gráfica, usada em modo gráfico como interface do jogo; o teclado, usado através de interrupções com recurso a rotinas escritas em Assembly, de forma a fazer a conseguir fazer a esolha do utilizador no menu como também movimentar os elementos do jogo, permitindo usufruir da jogabilidade. Quanto ao altifalante, permite-nos a inclusão de som no jogo.

Em relação ao nível de software, foi utilizado o editor de texto usado nas aulas práticas, *notepad*++, para editar código, para compilar e testar o jogo, como também o assemblador NASM, o compilador DJGPP, a makefile, o SVN para a gestão das versões do projecto.

O projecto foi desenvolvido no Sistema Operativo Windows 98, através de uma máquina virtual criada pelo software Virtual PC.

2. Descrição do programa

O projecto desenvolvido, designado "Arkanoid" baseou-se no jogo já existente Arkanoid. O jogador ao iniciar o jogo é apresentado o Menu Principal onde este poderá escolher uma das seguintes opções: Jogar, Pontuações ou Sair.

Interface:

Os elementos presentes na interface durante o decorrer do jogo são:

- Cenário;
- Nave;
- Número de vidas que cada jogador tem;
- Tempo decorrido desde o início do jogo;
- Pontuação total.

A disposição dos elementos referidos obedece às seguintes regras:

- O cenario de jogo encontra-se na parte esquerda do ecra(nave, blocos, esfera);
- Na parte direita do ecrã encontram-se os restantes elementos: número de vidas disponíveis, tempo decorrido e pontuação total de cada jogador.

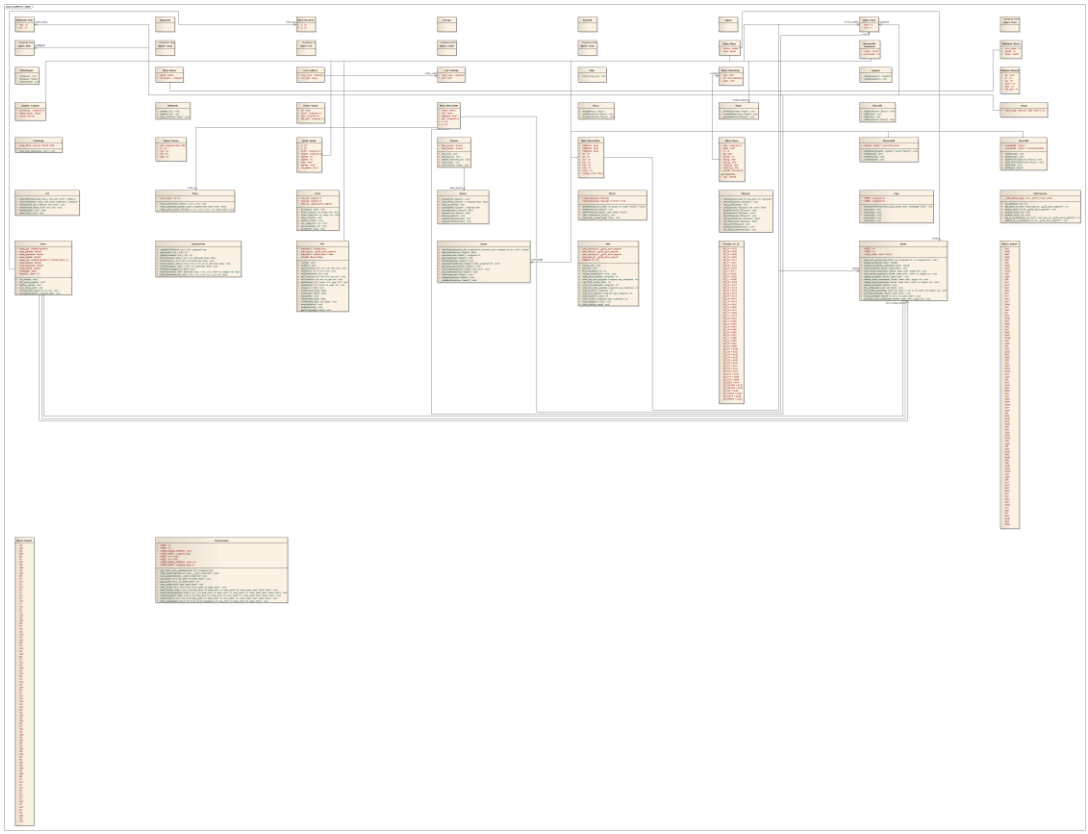
Regras de Jogo:

O Arkanoid é um jogo arcade que consiste em destruir uma série de blocos, usando para isso uma esfera que vai fazendo ricochete entre esses blocos, as paredes e uma "nave".

O objectivo do jogo é destruir todos os blocos de um dado mapa sem deixar a esfera sair do jogo pelo limite.

De forma a tornar o jogo mais viciante, o objectivo secundário será terminar o mapa no menor tempo e fazendo o máximo de pontuação possível.

3. Implementação



(UML presente na SVN)

3.1 Arquitectura do Programa

Segue-se a apresentação dos módulos implementados e uma breve descrição de cada um deles:

Timer - contém as funções que activam/desactivam as interrupções do TIMER.

Music - contém as funções necessárias ao uso do altifalante no programa, nomeadamente tocar notas/músicas no Menu Principal e no decorrer do jogo assim que ocorram colisões.

Altifalante - Módulo que será utilizado para reprodução de músicas/sons do jogo;

Teclado - Módulo que possibilitará o uso do teclado por parte do utilizador nos menus e decorrer do Jogo.

GraficaTexto – contém as funções necessárias para a utilização da placa gráfica em modo de texto. Usada na listagem das melhores pontuações.

GraficaVideo - contém as funções necessárias ao uso da placa gráfica do computador em modo gráfico. Incluí as funções de entrada/saída do modo gráfico, de desenho e obtenção de propriedades de pixéis, em suma, este módulo contém todas as funções responsáveis pela parte gráfica da aplicação;

Interrupcoes – está encarregue pelas interrupções, suporta todas as funções que permitem habilitar e inibir as mesmas, instalar e desinstalar os handlers respectivos.

List – Implementação simples de listas ligadas em C, contem funções de criar uma nova lista, de inserir elementos, de remover elementos e de libertar a memória da lista.

RTC – Contém funções necessárias à utilização do RTC, usado para a leitura dos dados do relógio, para os logs/records do jogo;

Bloco - Módulo que contém a estrutura de um "bloco" do jogo;

Mapa - Módulo que contém a estrutura dos mapas do jogo;

Jogo - Módulo que faz a gestão dos todos os eventos do jogo;

Jogador - Módulo que contém informação acerca do jogador do jogo(nome, mapa e pontução);

Sprite – compreende todas as funções relativas aos sprites utilizados durante o jogo, particularmente a função de leitura do xpm, as funções para criação das sprites e do desenho dos mesmos no ecrã, encontrando-se ainda neste módulo a função de animação das sprites.

Nave – providencia as funções de manipulação da nave do jogo.

Queue – contem funções de manipulação da fila constituída por caracteres;

GQueue – queue genérica equivalente ao modulo Queue, mas generalizado, ou seja aceita elementos de qualquer tipo. Esta queue é usada numa das implementação do rato, (a outra implementação usa variáveis voláteis, para passar os dados para as funções em C).

3.2 Funcionalidades

Funcionalidades implementadas:

- Sons ao longo do Jogo;
- Controlo de interrupções do teclado, do timer e do RTC;
- Sprites animados;
- Sprite (nave) controlada pelo teclado;
- Texto em modo gráfico;
- Pontuação em tempo real e no final do jogo;

Funcinalidades especificadas, mas não implementadas:

-Menu;
- Editor de níveis de jogo;
-Funcionamento do rato em conjunto com a nave do jogo;
-Ficheiro de texto que grava as melhores pontuações;
-Contagem da pontuação;

3.3. Detalhes Relevantes da Implementação

Foram criados vários módulos, e cada um tem a sua funcionalidade específica, e todas essas funcionalidades são reunidas no main/Jogo.

O tratamento de interrupções do teclado foi feito através da rotina escrita em assembly kbd_isr, e o rato através da rotina mouse_isr.

A contagem do tempo de jogo é feita no handler do RTC como as interrupções periodicas não estvam a acontecer utilizamos o get_time() e contadores auxiliares.

No módulo Music controla-se o momento em que os sons em execução devem ser silenciados.

3.4 Instruções de compilação e utilização

Para compilar o projecto basta fazer download do código-fonte, colocá-lo numa directoria à escolha, navegar até essa directoria, executar o comando "make", caso pretenda limpar os ficheiros anteriormente criados (ficheiros com extensão.o, executáveis e bibliotecas) basta executar "make clean".

Funcionamento do Programa

Para comprrender melhor o funcionamento do projecto, inclui-mos no SVN imagens e vídeos sobre o mesmo.

4. Conclusões

Este trabalho foi substancial para aprofundar e fortalecer conhecimentos de C, bem como para adquirir experiência e familiarização com programação de baixo nível e com acesso directo ao hardware.

Inicialmente foi feito uma planificação do trabalho, que não foi cumprido devido a dificuldades técnicas, agravadas por falta de meios para o desenvolvimento do projecto.

Se houvesse mais tempo, fariamos a implementação dos módulos que não foram implementados, uma optimização geral do programa e organizavamos melhor o código e a sua estrutura interna.

As maiores dificuldades sentidas foram, essencialmente, ao iniciar o projecto com as *sprites*, mais precisamente o uso de paletes personalizadas em conjugação com a leitura de ficheiros xpm e de seguida a localização da origem de page faults aquando do uso do rato.

Infelizmente, os resultados obtidos não foram tão bons quanto os que tinhamos idealizado.

5. Referências

DJGPP – http://www.delorie.com/djgpp/

NASM – http://www.nasm.us/

GCC - http://gcc.gnu.org/

Makefile – http://www.gnu.org/software/make/

SVN - http://subversion.tigris.org/

Microsoft Windows 98 – http://www.microsoft.com/en/us/default.aspx

Arkanoid - http://pt.wikipedia.org/wiki/Arkanoid

Anexos

Como o relatório ia ficar demasiado extenso e difícil de consultar decidimos por todos os anexos na SVN. Deste modo a consulta de comentários ao código, diagrama de funções e UML torna-se muito mais fácil.