Universidade do Porto

FEUP Faculdade de Engenharia

EIC0020

Laboratório de Computadores

Commodore

Turma 5, Grupo 4

Por:

Pedro Silva - ei11061

Rui Figueira – ei11021

Contents

Resumo:..... 3
Introdução 4
Descrição do Jogo: 5
Originalidade:..... 6
Explicação da Lógica do jogo 7
Arquitectura do Programa: 10
Detalhes relevantes da implementação 11
Principais Contribuições do Grupo 12
Conclusão:..... 12
Anexos e links usados: 13

Resumo:

Este relatório descreve o projecto final que foi desenvolvido na unidade curricular Laboratório de Computadores(EIC0020) e que teve como principal objectivo aprofundar o nosso conhecimento sobre a utilização da interface de "hardware" de alguns periféricos usuais de um PC, assim como combinar esse conhecimento com a utilização estruturada da linguagem de programação C.

Os periféricos utilizados neste projecto foram os seguintes:

- A placa gráfica, utilizada em modo gráfica,usada ao longo de todo jogo para a sua apresentação (menu, jogo, ecra final).
- O teclado, utilizado para selecionar a opçao desejada no menu inicial e para direcionar e selecionar o cursor in-game para a escolha do quadrado do tabuleiro.
- O timer, utilizada na limitação do tempo disponível para cada jogada feita pelo jogador a 20s e limite de escolha do menu principal a 60s.
- O RTC, utilizado para calcular o tempo do jogo.

As ferramentas de software utilizadas foram as seguintes:

- Eclipse, juntamente com o RSE (Remote System Explorer) plug-in para permitir desenvolver codigo num ambiente mais agradável de trabalhar do que na consola do Minix.
- Vmware Player para emulação do sistema operativo Minix 3.
- Sistema operativo Minix 3,sistema sobre o qual corre o nosso projecto e para o qual criamos as drivers ao longo das aulas práticas.
- SVN para o controlo dos ficheiros do projecto,mantendo informação relativa ás diferenças feitas de revision para revision.
- Doxygen para gerar a documentação relatica ao código.
- Enterprise Architect para gerar o diagrama UML relativo ás relações no nosso código.

Introdução

No âmbito da unidade curricular Laboratório de Computadores (EIC0020),foi-nos pedido a criação de um projecto que incorporasse os "drivers" estudados e implementados pelo grupo ao longo das várias aulas laboratoriais realizadas durante o semestre. Após a discussão de algumas ideias, decidimos que o projecto seria um jogo baseado no famoso jogo da "batalha naval". Uma vez que a nossa versão tinha como objectivo ser diferente da original, decidimos atribuir o nosso próprio nome á nossa versão - "Commodore", nome inspirado pela patente mais alta da marinha, que tem ao seu comando vários navios.

Inicialmente quando foi entregue a proposta e respectiva especificação do projecto, o docente da cadeira referiu algumas dificuldades que poderiamos encontrar na implementação simultânea da placa gráfica em modo de texto e em modo de video, pelo que alteramos o projecto para funcionar exclusivamente em modo de video. Foi-nos também indicado que a utilização do periférico serial port seria um trabalho bastante moroso pelo que decidimos tentar implementa-lo apenas se existisse tempo para tal.

Tal como já referimos na secção anterior,estamos a utilizar neste projecto a placa gráfica para fazer o *display* do nosso jogo, o teclado para navegar pelos menus e tabuleiro de jogo, fazendo a seleção do botao ao premir *Enter*, o *timer 2* para limitar o tempo de jogada do utilizador, o *RTC* para calcular o tempo de jogo.

Todo o projecto foi escrito usando o Eclipse em modo Remote System Explorer pois dava acesso a um ambiente de programação mais ergonómico e mais rapido que a linha de commandos do minix. A compilação foi realizada na maquina virtual vmware do minix disponibilizada pelo docente da cadeira.

Descrição do Jogo:

Como já foi referido o nosso projeto foi baseado no clássico jogo da Batalha Naval. Este jogo tem como objectivo afundar os navios do adversário antes que este consiga afundar os nossos.

Tal como no jogo original, o nosso jogo é um *turn-based game*, isto é, o jogador tem um tempo limite para realizar uma jogada (no nosso jogo definimos este tempo como 20 segundos) ao fim do qual será a vez do computador jogar. Este ciclo continua até que um dois participantes ganhe, ou seja, até que um dos dois consiga afundar todos os navios do oponente.

Ao inicializar o jogo, o utilizador depara-se com um "menu" com resolução 1024x768, no qual pode selecionar a opção "*Play*" ou a opção "*Quit*" usando o teclado como meio de navegação, nomeadamente as setas de direcção ou as teclas wasd. Ao selecionar a opção play este menu será apagado e substituido pelo ecran de jogo onde se encontram dois tabuleiros de jogo, um pertencente ao jogador onde o computador vai fazer os seus ataques e um correspondente ao computador onde o jogador vai realizar os seus. [O display deste ecrân é explicado em maior detalhe na secção "Originalidade"]. Tal como no menu anterior o jogador utilizará o teclado para selecionar qual o quadrado que pretende atacar. Mal o utilizador entre neste ecrân poderá verificar que os seus barcos (coloridos a cinzento no vmware) já se encontram posicionados no tabuleiro de defesa (no canto superior esquerdo), assim como os do AI (embora invisiveis ao utilizador). A primeira jogada é realizada pelo computador, fato esse, que se pode comprovar olhando para o "tabuleiro de defesa". As jogadas do utilizadores são limitadas por um tempo limite de jogo (20 segundos como já referido) ou até pressionar a tecla *Enter* num quadrado que ainda nao foi atingido (cuja cor nao seja ou vermelha ou azul escura). Quando o jogador ou o AI afundam um navio, o respectivo desenho do lado lateral do tabuleiro é colorido de vermelho escuro, permitindo assim saber quais os navios que já foram eliminados.

Este ciclo de jogo continua até o jogador ou o AI,afundar todos os navios do inimigo. Quando esta situação se verifica,o ecran de jogo é apagado e é substituido por outro que mostra um texto indicando ao utilizador se ganhou ou perdeu o jogo e o tempo de duração (cuja implementação a nivel de codigo utiliza o RTC) esperando 7 segundos ao fim do qual o jogo termina, retornando à linha de comandos do minix.

Originalidade:

As variações entre a nossa versão do jogo e a original são na forma como o jogo se processa. Incluímos no nosso jogo 3 ataques especiais:
Um ataque "*horizontal/vertical volley*" que varre uma linha (horizontal ou vertical), destruindo tudo o que nela se encontra. Premindo o botao 1 seleciona-se o ataque na horizontal, premindo o 2 seleciona-se o ataque vertical.

Um ataque "*scout*" que selecciona aleatoriamente um quadrado de um barco ainda por atingir. Este ataque nao realiza um hit no barco,mas sim,indica onde ele se encontra (daí ser numa cor laranja para se poder diferenciar).
Um ataque "bombardement" que destroi uma area de um quadrado 3x3,sendo o centro do ataque o quadrado que o utilizador seleciona.

Outro aspecto que decidimos mudar foi o *display* do jogo. Normalmente a batalha naval é jogada com dois tabuleiros colocados lado-a-lado mas nós inspiramo-nos numa versão original da batalha naval jogada com lápis e papel e optamos por uma visualização mais centrada do tabuleiro de ataque do utilizador.Este,por sua vez, possui um tamanho maior para realçar a sua importância, enquanto que o tabuleiro de "defesa" do jogador (aquele em que se encontram os seus barcos) fica no canto superior esquerdo com um tamanho mais reduzido. Os barcos de cada participante são os que se encontram ao lado do respetivo tabuleiro. Para o utilizador humano, os barcos que afundará (do inimigo) estarão ao lado do tabuleiro de ataque(o maior deles) enquanto que os seus navios encontram-se ao lado do tabuleiro mais pequeno (de defesa).

Explicação da Lógica do jogo

O Game-loop:

Tal como já foi referido,o commodore é um turn-based game e como tal, a nível de implementação, o que fizemos foi criar um game-loop que so termina quando ou o jogador ou o AI tenham destruido todos os barcos do adversário. O game-loop é constituido por uma jogada do jogador e por uma jogada do AI. Cada jogada do jogador tem um limite temporal de 20 segundos, ao fim dos quais,se não for executado nenhum ataque, o AI joga.

Os Tabuleiros:

Ao inicializar o jogo,surgem dois tabuleiros. Cada um possui o seu próprio tamanho. O tabuleiro no qual o AI joga possui quadrados de 28x28 pixeis, enquanto o do jogador tem quadrados de 35x35 pixeis. Cada tabuleiro é representado informaticamente por duas estruturas de dados. Uma estrutura de quadrados que por sua vez possuem informação relativamente ao seu conteudo e se ja foram premidos ao longo do jogo e uma outra estrutura que representa um tabuleiro de ID's que funciona como um mapa de onde estão os navios. Os tabuleiros estão pintados de um azul claro representando o mar,possuem um outline preto de modo a individualizar melhor os tabuleiros e ainda quadrados referentes á linha e á coluna com números e letras respectivamente de modo a melhorar a interface gráfica.

Os Hits/Miss:

Sempre que o jogador carrega no enter,ele está a simular o "ataque" a um quadrado. Este "ataque" pode originar um miss no caso de esse quadrado estar vazio ou um hit no caso de conter algo. Quando o jogador ou o AI originam um miss, o que acontece é que é desenhado uma sprite animada no sitio do ataque correspondente a um remoinho, acabando o quadrado por ficar azul escuro no final. No caso de ter acertado em algo é também desenhada uma sprite animada,mas representando uma explosão e o quadrado fica vermelho no final. Em ambos os casos a estrutura de dados composta pelos quadrados com informação sobre se ja foram premidos ou não, é actualizada de modo a impedir que o quadrado seja de novo acertado em turnos seguintes.

O AI:

Para fazermos o AI engine,decidimos fazer um AI fácil de vencer, que não tivesse em conta tabelas de probabilidade de localização de barcos nem que tivesse em conta quantos barcos ainda existem no tabuleiro de modo a poder melhorar os seus targets. Portanto, o nosso AI o que faz é em cada turno criar uma coordenada de ataque aleatorio no mapa que ainda não tinha sido

atacada. Quando acerta num barco, muda o seu modo para o modo de ataque enquanto não destruir o barco. Sempre que acerta num barco guarda essa informação numa estrutura e vai procurando nos quadrados adjacentes por mais "peças" desse mesmo barco. Quando destroi o barco,volta de novo ao modo de procura aleatório.

A disposição dos barcos:

Tanto os barcos do AI, como os do jogador são colocados de forma aleatória em cada tabuleiro. No inicio de cada jogo, o programa corre um algoritmo que garante que nenhum barco sai fora dos limites do tabuleiro ou que sobrepoe outro barco. De forma a tornar o jogo mais "challenging", fizemos o algoritmo de modo a não impedir que os barcos fiquem adjacentes, permitindo assim vários tipos de "display" possiveis. À medida que os barcos são desenhados no ecra, as respectivas estruturas de dados que possuem informação sobre o conteudo e os ID's dos tabuleiros são actualizadas.

O cursor do teclado:

De modo a permitir manter a informação sobre os movimentos do jogador com o teclado, criamos uma estrutura que representa informaticamente o cursor do teclado e que esta sempre a ser actualizada com a informação sempre que o jogador o mexe. Contém informação relativa á sua posição no tabuleiro e á medida que é mexido desenha um outline para indicar ao jogador em que posiçao está. Ao mesmo tempo apaga o outline da posiçao anterior. Essencialmente é o elemento principal da ligação do jogador ao jogo.

Os ataques especiais:

Cada ataque especial tem associado uma flag que representa se já foi ou não utilizado . Esta flag é activada quando é utilizado pela primeira vez o ataque especial, impedindo assim de serem feitos mais do que uma vez ao longo de um jogo. O jogador ao premir o botão de um ataque é verificado se a flag está "up" e se não estiver o ataque é realizado. O AI também pode realizar ataques especiais, sendo estes activados de forma aleatória com base no número de hits do jogador.

- **O Volley:**

Neste ataque especial,com base na posiçao actual do cursor do teclado e do botão premido identificando se é um "vertical volley" ou um "horizontal volley", o que é feito é que para todos os elementos dessa linha/coluna, são verificados todos os quadrados e actualizados no desenho com o respectivo output do ataque. A estrutura de dados do tabuleiro é também actualizada com base nos hits/miss.No caso de ser usado pelo o AI, se tiver sido encontrado

alguma peça de um navio,o AI muda para o modo de ataque com base nas coordenadas da última peça encontrada com o volley.

- **O Scout:**

Neste ataque especial, existe um algoritmo que procura em todo tabuleiro por um barco que ainda não tenha sido atingido nenhuma vez e quando o encontra simplesmente muda a cor do quadrado. No caso de ser usado pelo o AI, o AI muda para modo de ataque com base nas coordenadas desse quadrado descoberto.

- **O Bombardement:**

Neste ataque especial ,com base na posição actual do cursor do teclado é feito um ataque com tamanho de um quadrado 3x3 usando a posição do cursor como ponto médio. A estrutura de dados do tabuleiro é também actualizada com base nos hits/miss.

Os Barcos Laterais:

No inicio do jogo são desenhados barcos laterais para cada tabuleiro, identificativos dos barcos que residem nesse mesmo tabuleiro. Quando um dos barcos é acertado, existe um algoritmo que verifica se o barco com esse respectivo ID já foi destruido, se já o tiver sido, o algoritmo verifica qual o ID e desenha o respectivo barco lateral a vermelho de forma a identificar que ja foi destruido.

Arquitectura do Programa:

O principal ficheiro do nosso jogo é o commodore.c, sendo nele que se localizam todas as funções necessárias ao funcionamento do jogo,bem como todas as váriaveis globais que representam as estruturas importantes do nosso projecto. Embora seja neste ficheiro que o core do project se encontre, é necessária a ligação a vários outros ficheiros para permitir o funcionamento do jogo. Em baixo estão descritos os principais ficheiros e qual a sua função.

Pequena descrição dos principais ficheiros e das suas funcionalidades:

- main.c - Inicializa o jogo,fazendo o sef_startup(),permitindo que jogo corra como um "processo privilegiado" de modo a ter acceso directo aos recursos de hardware e chama a função que dá inicio ao jogo.
- commodore.c - Contém todas as funções relativas ao funcionamento da lógica do jogo (AI engine,ataques especiais, logica dos turnos,posicionamento dos barcos, movimento do cursor). Contém ainda todas as funções de desenho da interface gráfica.
- interupts.c - Contém as funções responsáveis pela instalação e desinstalação das interrupções usadas e o interrupt handler para os periféricos.
- pixmap.h - Contém todos os XPM de todas as sprites usadas no jogo.
- rtc.c - Contém as funçoes de leitura e tratamento de informação do RTC, assim como as funçoes de subscrição do mesmo. Possui ainda as funções de desenho do tempo do jogo.
- kbd.c - Contém as função de leitura de scancodes do teclado, assim como as funções de subscrição do mesmo.
- timer.c - Possui as funções que activam e desactivam as interrupções do Timer 2.
- video_gr.c - Possui as funções necessárias para o uso da placa gráfica do computador em modo gráfico, bem como todas as funçoes auxiliares ao desenho da parte gráfica do nosso jogo.

Detalhes relevantes da implementação

Em relação à implementação dos conhecimentos adquiridos nesta cadeira ao longo do semestre, acreditamos que o grupo soube usar de forma adequada os drivers que efetivamente implementou. Demonstramos saber usar o teclado de forma eficaz sem qualquer problema,conseguindo também a nível do modo de video,usa-lo sem a necessidade de qualquer tipo de auxilio gráfico como double buffering,vertical synchronization ou page flipping. Achamos também que a nossa implementação do real-time clock driver demonstra bons conhecimentos adquiridos pois não foi necessário subscrever ao mesmo para tratar da sua informação,podendo aceder diretamente aos registos para obter o seu conteudo e trabalhar diretamente nele.

A nossa versão final do jogo possui algumas diferenças comparativamente á nossa especificação inicial. Ao contrário do que tinhamos planeado,acabamos por não ter tempo para implementar o serial port e como tal acabamos por fazer um jogo sem multiplayer. Para contrabalançar essa perda fizemos um AI engine que permitisse ao utilizador jogar contra o computador. Na especificação inicial tinhamos também definido como periférico padrão a ser utilizado in-game , o rato, e embora tenhamos conseguido meter o rato a funcionar com o jogo, devido a sérias dificuldades em conseguir fazer com que o rato seguisse a trajéctória pretendida, acabamos por implementar o teclado como periférico padrão. Apesar de termos o speaker totalmente funcional, como não pudemos testar os sons que fizemos com o jogo devido a só termos feito essa parte na última semana, semana essa em que os laboratórios estiveram fechados, optamos por não arriscar colocar a parte do speaker no projecto enviado. No que toca ás restantes especificações, pensamos que cumprimos com todos os objectivos, tendo implementado o jogo em modo gráfico, usando as sprites que tinhamos planeado, criado o menu inicial ,metendo as interrupções do teclado a funcionar juntamento com o timer e aplicando toda a lógica de jogo que tinhamos idealizado. Acabamos ainda por implementar mais algumas novas funcionalidades que não estavam especicificadas como 3 ataques especiais ,a utilização do RTC para calcular o tempo do jogo, a disponibilização gráfica de que barcos ja foram destruidos e um menu de final de jogo.

Principais Contribuições do Grupo

Pedro Silva:

- Handler do RTC.
- Sprites do menu inicial, jogo e menu final.
- Ligação entre todos os módulos dos periféricos e interrupções.

Rui Figueira:

- Handler do Teclado.
- Toda a Lógica do Jogo.

Conclusão:

Desenvolver este projecto permitiu-nos obter várias ferramentas úteis a serem usadas para a realização de projectos de maior nível enquanto futuros engenheiros informáticos. Entre essas ferramentas, lista-se o uso do sistema de sub-versões (SVN) para manter um histórico das alterações do nosso código, o lidar e corrigir bugs a um nível muito maior ao que estavámos habituados e a preocupação com a estruturação, modulação, optimização e documentação de todo o código escrito de forma a facilitar a sua referência. Para além destas vantagens a nivel de desenvolvimento de código, pudemos expandir o nosso conhecimento sobre a escrita de software ao nível de hardware e perceber melhor como os periféricos que utilizamos no nosso dia-a-dia funcionam. Tivemos ainda a oportunidade de melhorar a utilização estruturada da linguagem C.

Ao longo do desenvolvimento do projecto encontramos algumas dificuldades. Fora da implementação do código, como maior factor adverso, destacamos o tempo, que infelizmente não nos permitiu implementar o serial port como desejavamos. No que toca a escrita de código, o nosso maior problema foi por um lado conseguir perceber onde é que estavam os erros que estavam a fazer o programa crashar ou comportar-se de forma diferente do esperado e por outro conseguir lidar com alguns bugs relativos aos handlers dos periféricos que por vezes dependiam e tinham de afectar variáveis de outros ficheiros. Ambas as dificuldades essencialmente reforçam a falta de um debugger que nos permitisse ver o estado das variavéis ao longo da execução do código para permitir ver onde estavam os problemas.

Resumindo, achamos que o projecto foi uma boa maneira de colocar os conhecimentos aprendidos ao longo do semestre em prática. Não foi fácil conseguir colocar tudo direito e ter um jogo funcional, mas foi essa mesma dificuldade que tornou tão recompensador terminar o projecto.Se tivessimos tido mais tempo, teriamos implementado o serial port de modo a poder expandir a jogabilidade do nosso jogo ao multiplayer e teriamos tentado corrigir a trajectoria do rato de modo a possibilitar a sua utilização sem problemas durante o jogo.

Anexos e links usados:

Links usados:
Para a realização deste projecto utilizamos os seguintes sites para as mais variadas operações:

- http://www.convertmyimage.com/ conversão de bmp->xpm
- http://stackoverflow.com/ retirar duvidas sobre codigo
- http://cplusplus.com/reference/ procurar funções de stl de c que auxiliam-se o projecto
- http://web.fe.up.pt/~pfs/aulas/lcom2012/ e links neste site contidos com informações sobre a implementação de drivers e código fonte disponiblizado pelo docente da cadeira.
- https://svn.fe.up.pt/repos/lcom1213-t5g4 para guardar o nosso codigo durante o semestre
- http://faculty.qu.edu.qa/rriley/cmpt507/minix/ para visualização da documentação do minix

Para ver a especificação do relatório ou o diagrama UML do projecto por favor veja os documentos em anexo a este relatório.

Para fazer um download do projecto completo incluíndo este relatório, diagrama UML, especificação, documentação e codigo fonte por favor clique no link abaixo:

https://svn.fe.up.pt/repos/lcom1213-t5g4