



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Ano Letivo 2016/2017

2 de Janeiro de 2017

Engenharia Informática e da Computação (MIEIC)

Laboratório de Computadores

“Plunder Of The Vast Ocean”

Grupo T5G07

Rui André Rebolo Fernandes Leixo

up201504818@fe.up.pt

Nuno Manuel Ferreira Corte-Real

up201405158@fe.up.pt

Introdução

Neste projeto, no âmbito da cadeira Laboratório de Computadores do curso MIEIC da FEUP, foi desenvolvido o jogo “Plunder of the Vast Ocean”, um jogo inspirado por vários jogos da categoria *top-view shooter*, mas apresentando um tema marítimo, inspirado pelo célebre jogo “Batalha Naval”.

É um jogo simples e intuitivo de aprender e jogar, com objetivos e jogabilidade claramente definidos, mas com potencial para implementar as diferentes mecânicas aprendidas nas aulas de LCOM e não demasiado simples de desenvolver.

Descrição do jogo

O jogador controla um barco. Este último pode ser movido para a frente, parar e rodar para os dois lados. Ao avançar, o jogador tem à sua disponibilidade duas velocidades: a velocidade genérica e a velocidade rápida.

Para atacar, o barco possui dois canhões laterais. Cada vez que um canhão é disparado há um período de *cooldown* que o jogador tem que esperar antes de poder voltar a disparar esse mesmo canhão. Dest modo, um dos desafios do jogo é a movimentação e posicionamento estratégico do barco de modo a esquivar-se dos inimigos e acertar neles com as balas dos canhões.

No mapa de jogo existem obstáculos (rochedos que o jogador tem que evitar, perdendo vida se colidir contra eles) e inimigos (monstros aquáticos que se deslocam aleatoriamente, com o objetivo de derrotar o jogador).

O objetivo do jogo é derrotar o maior número de inimigos possível. Estes aparecem aleatoriamente ao longo do jogo, devendo o jogador eliminá-los o mais depressa possível, caso contrário tornar-se-á progressivamente mais difícil sobreviver.

O jogo termina quando a vida do jogador desce até zero. O barco possui 100 pontos de *HP*. O jogador pode perder *HP* das seguintes maneiras:

- O jogador atropela uma cobra, perdendo 20 pontos de vida, matando a cobra;
- O jogador avança contra um rochedo, perdendo 10 pontos de vida por cada segundo que fica a tocar no rochedo;

Uma cobra possui 100 pontos de vida. Uma bala de canhão retira 50 pontos de vida a uma cobra.

Instruções de utilização

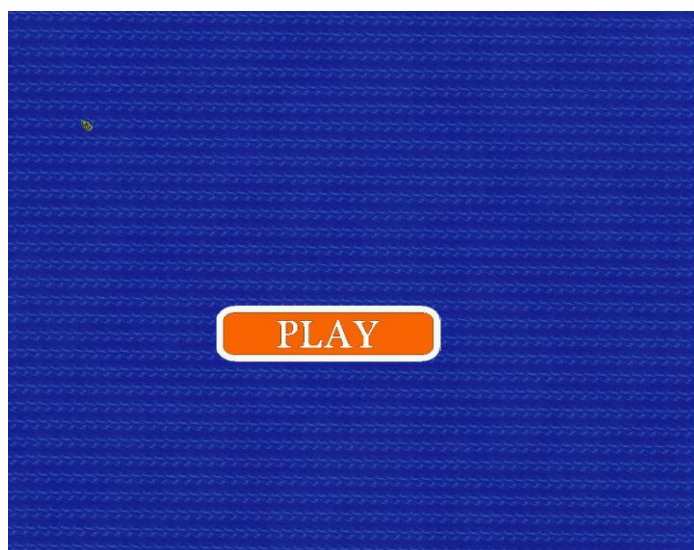
1.1 Setup do jogo

1. Fazer *checkout* ou *download* dos ficheiros do repositório do projeto
2. Navegar para a pasta raiz do projeto (pasta chamada *proj*)
3. Executar o comando **sh install.sh** (copia a *config* file para */etc/system.conf*, dá permissões de acesso ao serviço, copias os *sprites* para o caminho correto)
4. Navegar para a pasta *src* (*proj/src*)
5. Executar o comando **make**
6. Executar o comando **service run `pwd`/proj**

1.2 Controlos do jogo

O jogo apresenta uma mecânica relativamente simples. O jogador controla o movimento do baco com o teclado, e os canhões com os botões do rato.

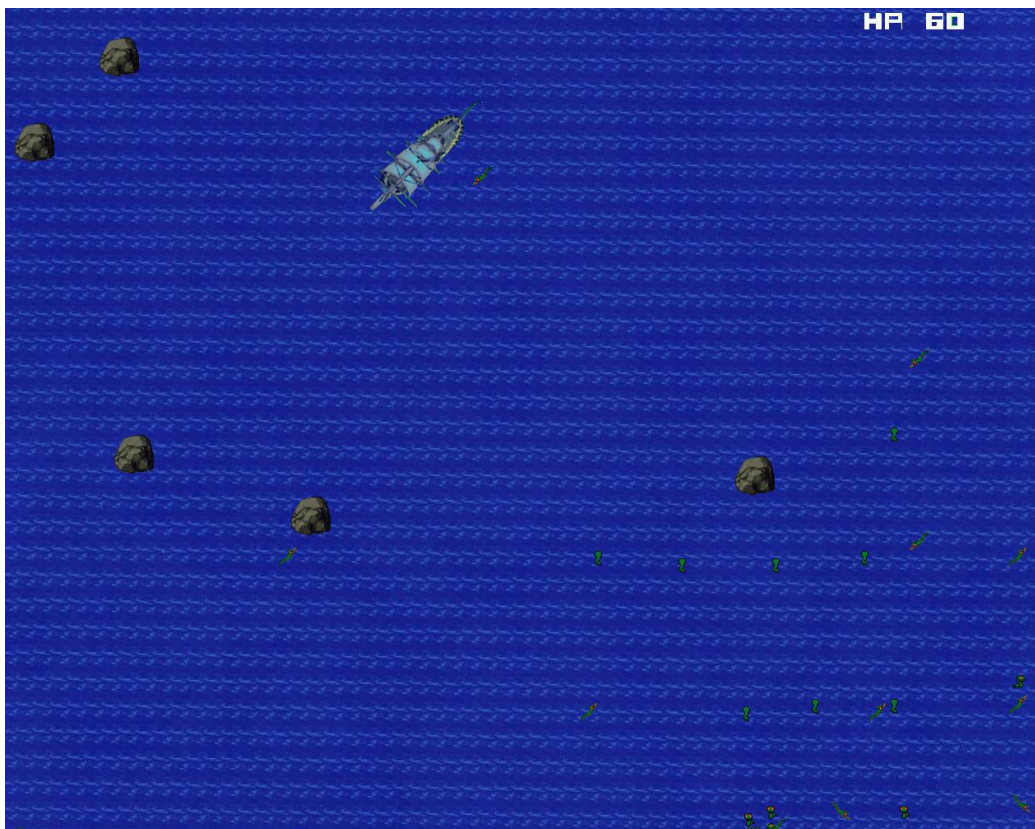
No menu inicial, o utilizador controla o rato, e utiliza o botão do lado direito para clicar no botão PLAY, dando assim início ao jogo.



Menu Inicial do jogo; o utilizador controla o cursor com o rato, dando início ao jogo quando carrega no botão play

No modo de jogo, as ações que cada tecla realiza são as seguintes:

W	Incrementa velocidade do barco (velocidade regular ou velocidade acentuada)
A	Roda o barco para a esquerda
D	Roda o barco para a direita
S	Decrementa a velocidade do barco. Se o barco estiver na velocidade regular, imobiliza-o na sua posição atual
Right-Click	Dispara os canhões direitos do barco
Left-Click	Dispara os canhões esquerdos do barco
Esc	Termina o jogo

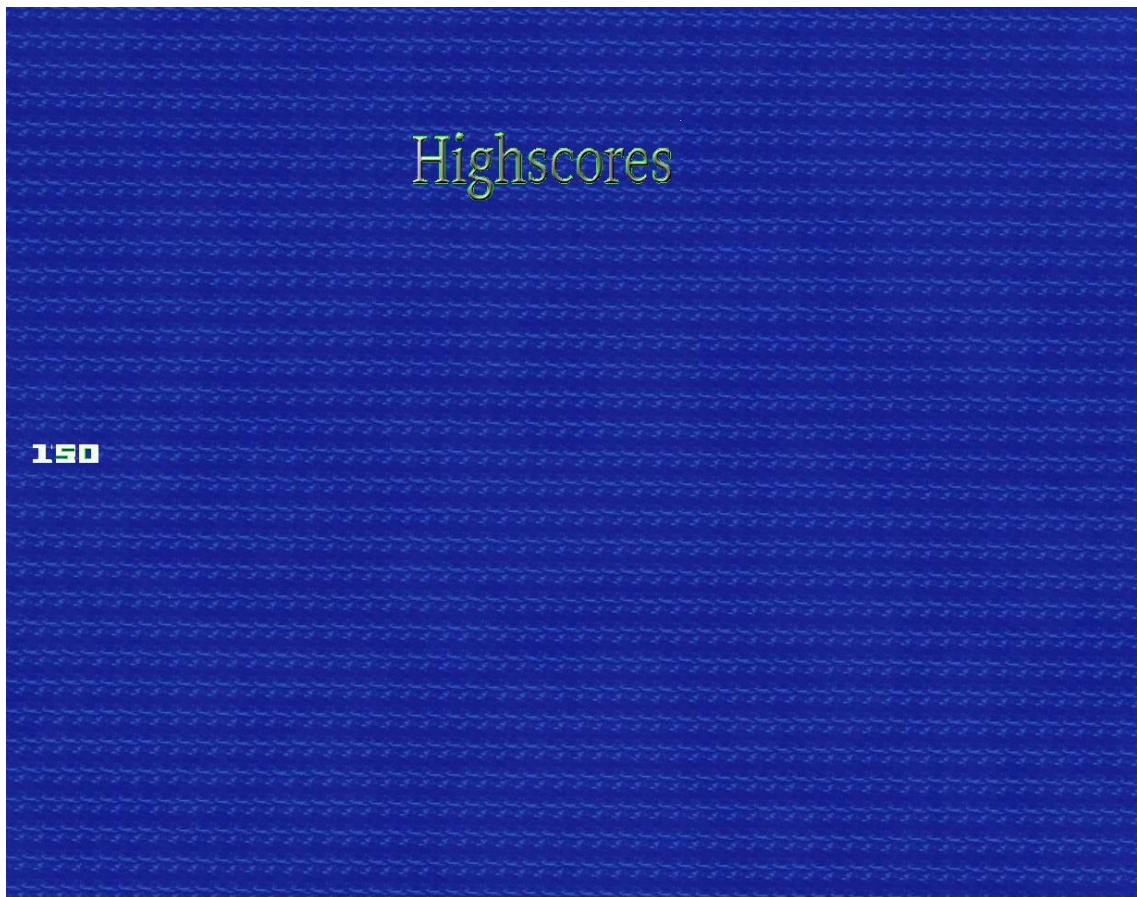


Estado de jogo: o barco, controlado pelo jogador, pode avançar nas direções N, S, E, W, NW, NE, SW, SE; os obstáculos, rochedos estáticos; os inimigos, cobras que se movem aleatoriamente, nas mesmas direções do barco; A vida do utilizador aparece a branco no canto superior direito

No final de jogo, quando o jogador fica sem vida ou carrega na tecla *Esc*, o programa sai do estado de jogo e entra no estado final, onde poderá ser visualizado a pontuação obtida pelo jogador. Este menu fica ativo durante 10 segundos; após este período o programa termina.

A pontuação do jogador sobe das seguintes maneiras:

- A cada segundo de jogo que passa o jogador recebe 10 pontos
- Por cada cobra que o jogador derrota com uma bala de canhão, recebe 100 pontos



Estado do projeto

Neste ponto analisamos o estado final do projeto e comparamo-lo com os objetivos previamente definidos no primeiro relatório, para cada dispositivo estudado na cadeira.

Timer 0

Este dispositivo é usado no âmbito da componente gráfica do jogo, mais precisamente para a contagem dos *fps's* (*frames per second*) , e na mecânica de jogo, servindo como contador do *cooldown* de cada canhão, após terem disparado. Cumpre, portanto, os objetivos que inicialmente foram definidos para o dispositivo.

Teclado

Este dispositivo é imprescindível para a mecânica de jogo do projeto, estando esta dependente do teclado. É, portanto, usado para a movimentação do barco. Inicialmente considerou-se utilizar este dispositivo como controlo dos canhões do navio; no entanto, devido a problemas encontrados com o rato, o teclado adquiriu uma função diferente. Apesar desta mudança, consideramos que o dispositivo cumpre os objetivos que inicialmente foram definidos para o dispositivo.

Rato

É usado como controlo dos canhões do barco, permitindo ao jogador disparar lateralmente balas de canhão, usadas para derrotar os inimigos no mapa. No relatório inicial do projeto, esperava-se utilizar este dispositivo para controlar o barco, objetivo este que não foi possível de realizar. No entanto, o rato continua a possuir uma função essencial à mecânica de jogo, considerando-se, portanto, que Cumpre parcialmente os objetivos que foram definidos para o dispositivo.

Porta de Série

Dispositivo não utilizado no projeto. No planeamento inicial do jogo, foi considerada a possibilidade de introduzir um modo de jogo para dois jogadores, onde dois jogadores, cada um a jogar numa máquina diferente, partilhariam um mapa de jogo, podendo colaborar para derrotar o maior número de inimigos possível ou tentar derrotarem-se um ao outro. No

entanto, devido a outros problemas encontrados e a falta de tempo, não foi possível implementar esta *feature*.

RTC

Dispositivo não utilizado no projeto. Foi considerada a possibilidade de implementar uma funcionalidade na qual quando o jogo terminasse, um menu *Game Over* surgiria, apresentando ao jogador a data atual (obtida através do RTC) e os *high-scores* atuais. No entanto, devido a outros problemas encontrados e a falta de tempo, não foi possível implementar esta *feature*.

Placa Gráfica

Utilizado na visualização dos gráficos de jogo, na construção de um *triple-buffer* (no qual um *buffer* é encarregado de imprimir o mar, outro imprime as restantes componentes de jogo no seu estado atual e o último serve como *next-frame buffer*, carregando as restantes componentes de jogo no estado seguinte, prontas a imprimir) de modo a permitir uma visualização mais suave, sem *screen flickering*.

Neste projeto, foi usado o modo gráfico 0x11A, de 16 Bits, RGB 5:6:5, 64 mil cores.

Organização e estrutura do código

Neste tópico, é especificada a estrutura do jogo tendo em conta os vários ficheiros que constituem o código do projeto:

timer.c

Contém as funções *subscribe*, *unsubscribe*, e de geração de um determinado período de tempo, usadas na componente gráfica do projeto e em parte da jogabilidade. Adicionalmente, possui as funções desenvolvidas nas aulas laboratoriais relativas ao periférico.

mouse.c

Contém as funções *subscribe*, *unsubscribe*, e de processamento das interrupções e dos *packets* do rato. Adicionalmente, possui as funções desenvolvidas nas aulas laboratoriais relativas ao periférico.

kbd.c

Contém as funções *subscribe*, *unsubscribe*, e de processamento das interrupções dos scancodes do teclado. Adicionalmente, possui as funções desenvolvidas nas aulas laboratoriais relativas ao periférico.

game.c

Contém todas as funções relacionadas com a lógica de jogo, o seu início, atualização e fim.

boat.c

Contém todas as funções relacionadas com o barco: sua criação, movimentação, impressão, processamento de colisões, destruição, processamento do estado atual.

enemy.c

Contém todas as funções relacionadas com o inimigo: sua criação (de modo aleatório), movimentação, impressão, processamento de colisões, destruição, processamento do estado atual.

obstacle.c

Contém todas as funções relacionadas com os obstáculos do mapa: sua criação (de modo aleatório), impressão e destruição.

cannonball.c

Contém todas as funções relacionadas com as bolas de canhão que o barco dispara. Sua criação, impressão, processamento de colisões, destruição.

vbe.c

Contém funções necessárias à leitura de informação do modo gráfico. Adicionalmente, possui todas as funções desenvolvidas nas aulas laboratoriais relativas ao periférico.

bitmap.c

Código retirado ao aluno Henrique Ferrolho e adaptado ao projeto e modo de vídeo utilizado, contendo as funções necessárias à impressão de bitmaps no ecrã.

video_gr.c

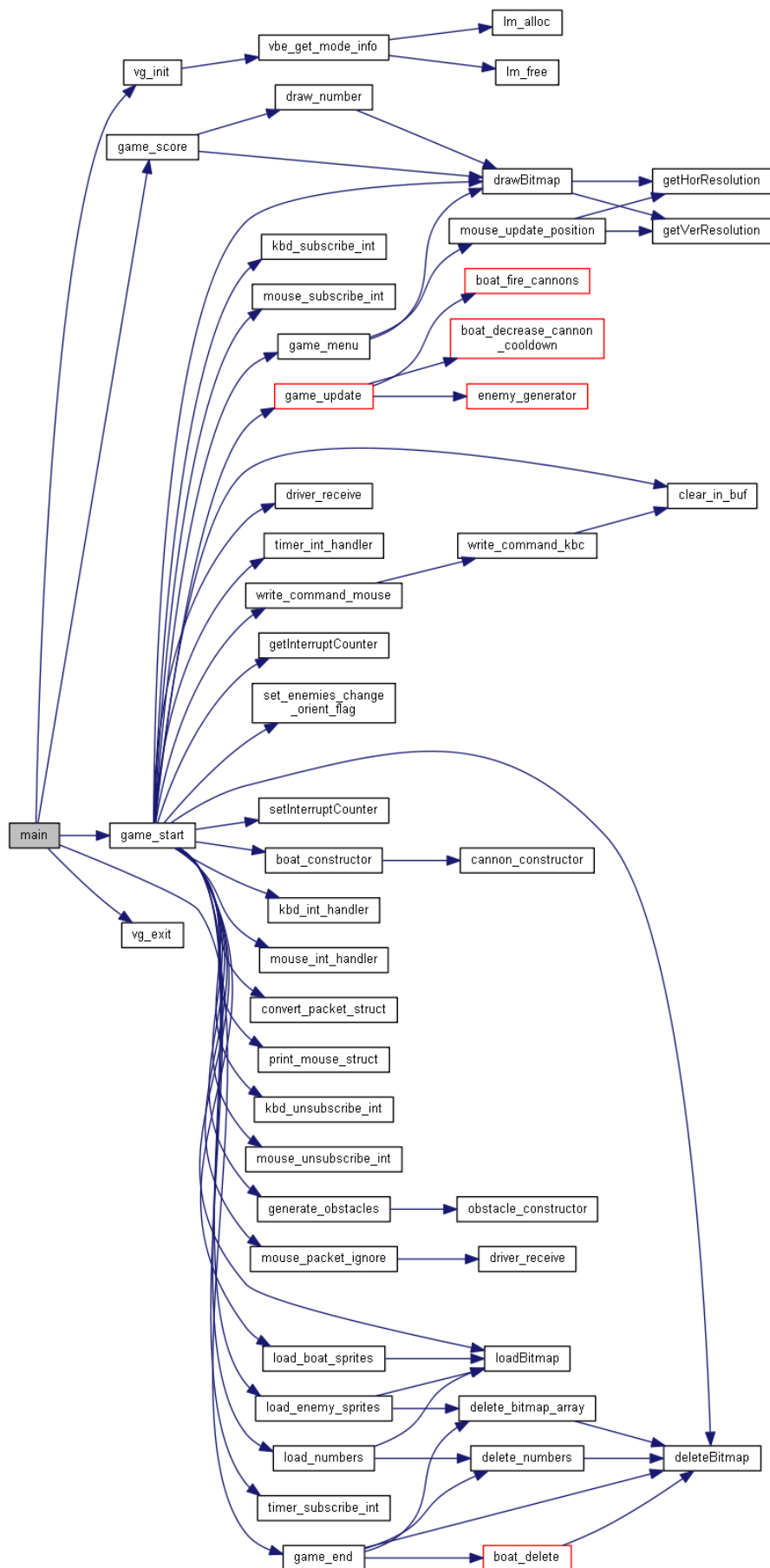
Código fornecido nas aulas e adaptado ao projeto, contendo as funções necessárias à inicialização e terminação do modo gráfico.

proj.c

main do projeto. Inicializa o modo gráfico, efetua as subscrições e des-subscrições necessárias à inicialização e terminação do programa, respetivamente, e relativamente aos diferentes periféricos utilizados.

É relevante mencionar que a Inavegação pela lógica do jogo baseia-se numa máquina de estados, constituída por 3 estados: Menu Inicial, Estado de Jogo, Menu Final.

Gráfico de chamada de funções



Distribuição do trabalho

Os diferentes módulos foram distribuídos da seguinte forma:

Kbd: Nuno

Mouse: Rui

Graphics: Rui

Timer: Nuno

A lógica de jogo foi desenvolvida em grupo pelos dois alunos.

Dificuldades encontradas

Durante o processo de desenvolvimento do jogo, os desafios maiores foram o cálculo das colisões entre todos os elementos do jogo (barco, obstáculos e inimigos), a procura de soluções que tornassem a lógica do jogo mais “leve” e eficaz.