



universidade de aveiro

escola superior de tecnologia e
gestão de águeda

PROJETO TEMÁTICO DE INFORMÁTICA APLICADA

Jogo de Batalha Naval

Relatório

Grupo GR27:

Carlos Azinhaga - 69468

Filipe Costa - 69363

Paulo Silva - 62201

Pedro Marquinhos - 70445

1º semestre 2012/2013



universidade de aveiro

escola superior de tecnologia e
gestão de águeda

PROJETO TEMÁTICO DE INFORMÁTICA APLICADA

Jogo de Batalha Naval

Relatório

Grupo GR27:

Carlos Azinhaga - 69468

Filipe Costa - 69363

Paulo Silva - 62201

Pedro Marquinhos - 70445

Orientador do Projeto:

Ciro Martins

1º semestre 2012/2013

Resumo

Este relatório descreve todo o trabalho realizado ao longo da elaboração do projeto no âmbito do módulo do Projeto Temático de Informática Aplicada. O relatório detalha os métodos utilizados para a definição dos objetivos, as divisões de tarefas, o desenvolvimento do projeto e, por consequência, os resultados obtidos.

O projeto consiste no desenvolvimento de uma aplicação em linha de comandos, em linguagem C, do tradicional Jogo da Batalha Naval. Nele são possíveis três configurações: escolher vários tamanhos de tabuleiros, que variam consoante a dificuldade; posicionar a frota automaticamente ou manualmente; e utilizar um modo de jogo tradicional ou modernista.

Em primeiro lugar, definiram-se os parâmetros principais a realizar e outros opcionais. Mais tarde, foram levantados os requisitos da aplicação.

Seguidamente, fez-se o planeamento e divisão das tarefas necessárias à conclusão do projeto.

Após as fases mais teóricas estarem concluídas passou-se para o desenvolvimento do código. Nesta etapa do projeto, deparámo-nos com algumas adversidades que nos obrigaram a fazer bastantes alterações. No entanto, conseguimos, através de vários testes, promover a concretização de alguns objetivos definidos inicialmente e ainda a verificação de possíveis falhas na aplicação.

Na fase final, foi elaborado este relatório e apresentada uma conclusão sobre o projeto e algumas perspetivas sobre desenvolvimentos futuros.

Índice

1 – Introdução	7
1.1 - Enquadramento do tema e objetivos	7
1.2 - Metodologia adotada.....	7
1.3 - Organização do relatório.....	9
2.1 - Levantamento de requisitos	10
2.2 - Definição de tarefas e atividades	12
2.3 - Divisão de tarefas.....	13
2.4 - Calendarização de tarefas.....	14
2.5 - Recursos utilizados.....	14
3- Descrição das atividades desenvolvidas	14
3.1 - Apresentação da solução desenvolvida	14
3.1.1 - Interface gráfica	14
3.1.2 - Descrição do programa	16
3.2 - Descrição do código	19
3.3 - Testes efetuados	22
5 – Conclusão	25

Índice de Figuras

Figura 1 - Metodologia adotada	8
Figura 2 - Fluxograma da aplicação	11
Figura 3 – Posicionamento manual da frota.....	15
Figura 4 – Posicionamento manual.....	17
Figura 5 –Tabuleiro de jogo.....	20
Figura 6 – Tabuleiro com duas jogadas.....	20

Índice de Tabelas

Tabela 1 - Definição das tarefas	12
Tabela 2 - Divisão das tarefas	13
Tabela 3 - Calendarização das tarefas	14
Tabela 4 - Simbologia utilizada para o tabuleiro	15
Tabela 5 - Simbologia utilizada para a frota	16
Tabela 6 - Simbologia utilizada para deteção de impacto	16
Tabela 7 - Requisitos concluídos	23

Lista de Siglas e Abreviaturas

ASCII - American Standard Code for Information Interchange.

1 – INTRODUÇÃO

Na introdução serão abordados tópicos como o enquadramento do tema e objetivos, a metodologia adotada e a organização interna do relatório.

1.1 - Enquadramento do tema e objetivos

O tema do projeto é "Jogo da Batalha Naval". O projeto consiste no desenvolvimento de uma aplicação informática em linguagem C, em que seja possível uma plataforma jogador contra jogador. O tamanho do tabuleiro de jogo varia consoante a dificuldade escolhida. Ainda é possível configurar a posição da frota de forma automática ou manual, sendo que na manual o jogador tem de posicionar um barco de cada vez, inserindo a coordenada inicial e a direção (vertical ou horizontal).

Este projeto integra-se no módulo do Projeto Temático de Informática Aplicada, do qual fazem parte as disciplinas "Informática e Programação" e "Metodologia e Gestão de Projeto". O principal objetivo deste projeto é a aplicação dos conhecimentos adquiridos nestas disciplinas, tais como:

- Desenvolvimento de um algoritmo estruturado;
- Implementação de um programa em linguagem C;
- Capacidade de detetar erros e corrigi-los;
- Planeamento e desenvolvimento de aplicações informáticas em grupo;
- Apresentação e documentação correta do projeto.

1.2 - Metodologia adotada

Nesta secção abordou-se a metodologia adotada para a execução do projeto e, também, as diferentes fases do seu desenvolvimento e suas conclusões.

A metodologia adotada é constituída por seis fases:

- Definição dos parâmetros/objetivos;
- Análise de requisitos;
- Planificação;
- Desenvolvimento de código;
- Fase de testes;
- Conclusões.

O seguinte esquema descreve as diferentes fases deste projeto e metodologia adotada.

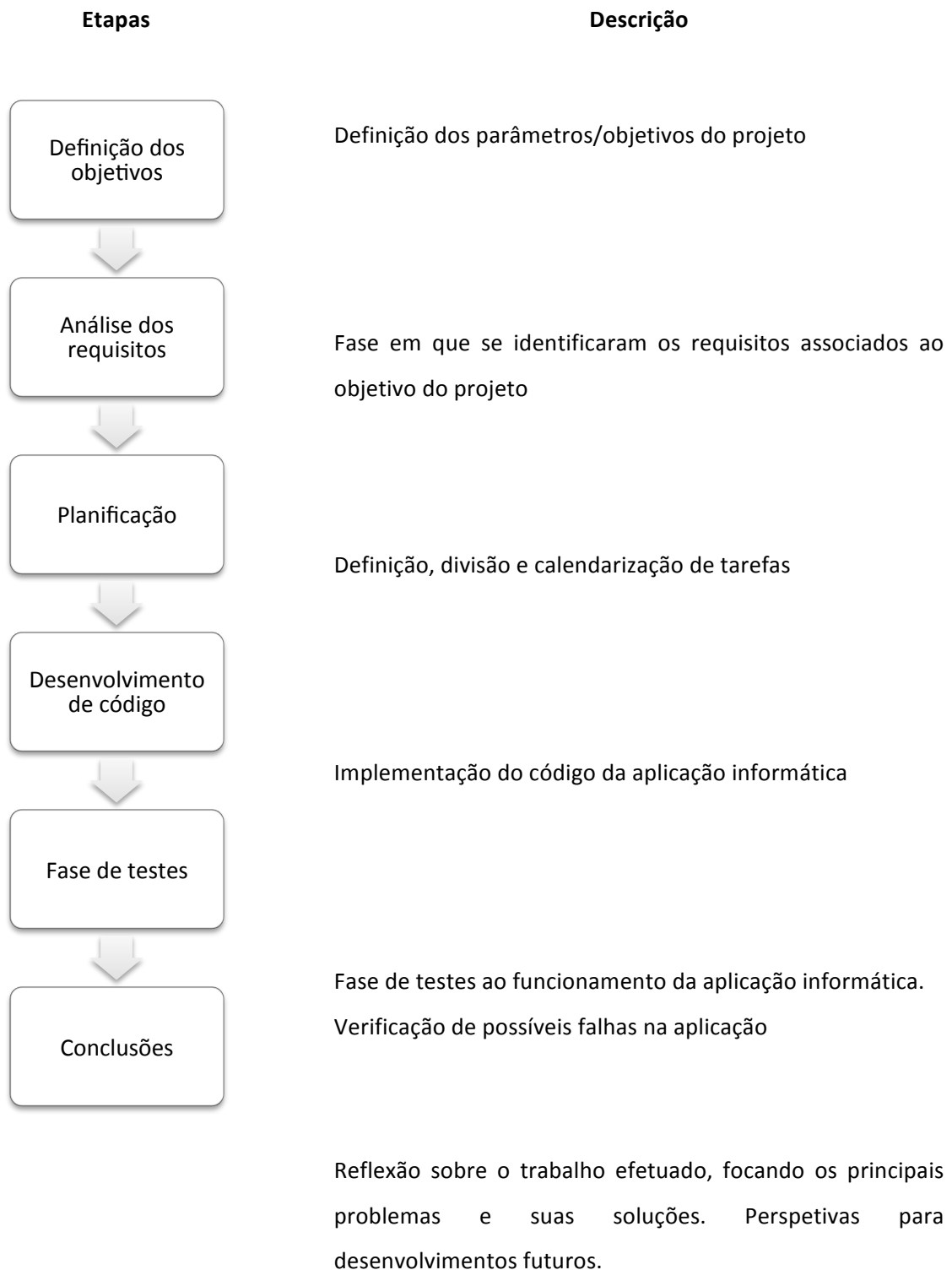


Figura 1 - Metodologia adotada

Foram realizadas várias reuniões de grupo com o orientador de projeto com o objetivo de supervisionar o trabalho, dar sugestões e esclarecimento de dúvidas.

1.3 - Organização do relatório

Este relatório está dividido em cinco capítulos principais, onde é descrito todo o trabalho desenvolvido na realização deste projeto.

- Capítulo 1 - Introdução: apresentação do tema do projeto e seu enquadramento, definição de parâmetros/objetivos, metodologia adotada e organização interna do relatório;
- Capítulo 2 - Planeamento do trabalho: descrição do levantamento dos requisitos, definição e divisão de tarefas e sua calendarização;
- Capítulo 3 - Descrição das atividades desenvolvidas: apresentação das soluções desenvolvidas, explicação do algoritmo e testes efetuados;
- Capítulo 4 - Análise de resultados: apresentação dos resultados do projeto;
- Capítulo 5 - Conclusões: descrição das principais conclusões do projeto e algumas perspetivas sobre desenvolvimentos futuros.

2 - PLANEAMENTO DO TRABALHO

O planeamento do trabalho divide-se em cinco tópicos: levantamento de requisitos, definição de tarefas e atividades, divisão de tarefas, a calendarização de tarefas e os recursos utilizados.

2.1 - Levantamento de requisitos

O levantamento de requisitos foi efetuado com base no enunciado do projeto e refere-se a algumas funcionalidades que a aplicação deve ter, como por exemplo:

- O jogo deve ter uma representação visual. Para tal é desenhado um tabuleiro com $M \times M$ posições;
- Funcionalidade de Jogador contra Jogador;
- Efetuar a deteção do local de impacto (água ou algum barco);
- Fim do jogo. O Jogo termina quando uma das frotas estiver destruída.

Para além das funcionalidades exigidas no enunciado foram adicionadas outras, como por exemplo:

- A dimensão do tabuleiro varia consoante a dificuldade escolhida:
 - Fácil: 10x10;
 - Normal: 12x12;
 - Difícil: 15x15.
 - O posicionamento da frota que pode ser automático ou manual:
 - Automático: a frota é posicionada de forma autónoma;
 - Manual: pergunta-se a orientação do barco (vertical ou horizontal) e depois a primeira coordenada.
- É efetuada validação de dados com o âmbito de não haver sobreposição de barcos.
- Quando um barco é atingido o computador emite um bip.

Com base em todos os requisitos elaborou-se um fluxograma representado na figura seguinte. Em suma é o algoritmo de suporte ao desenvolvimento da aplicação.

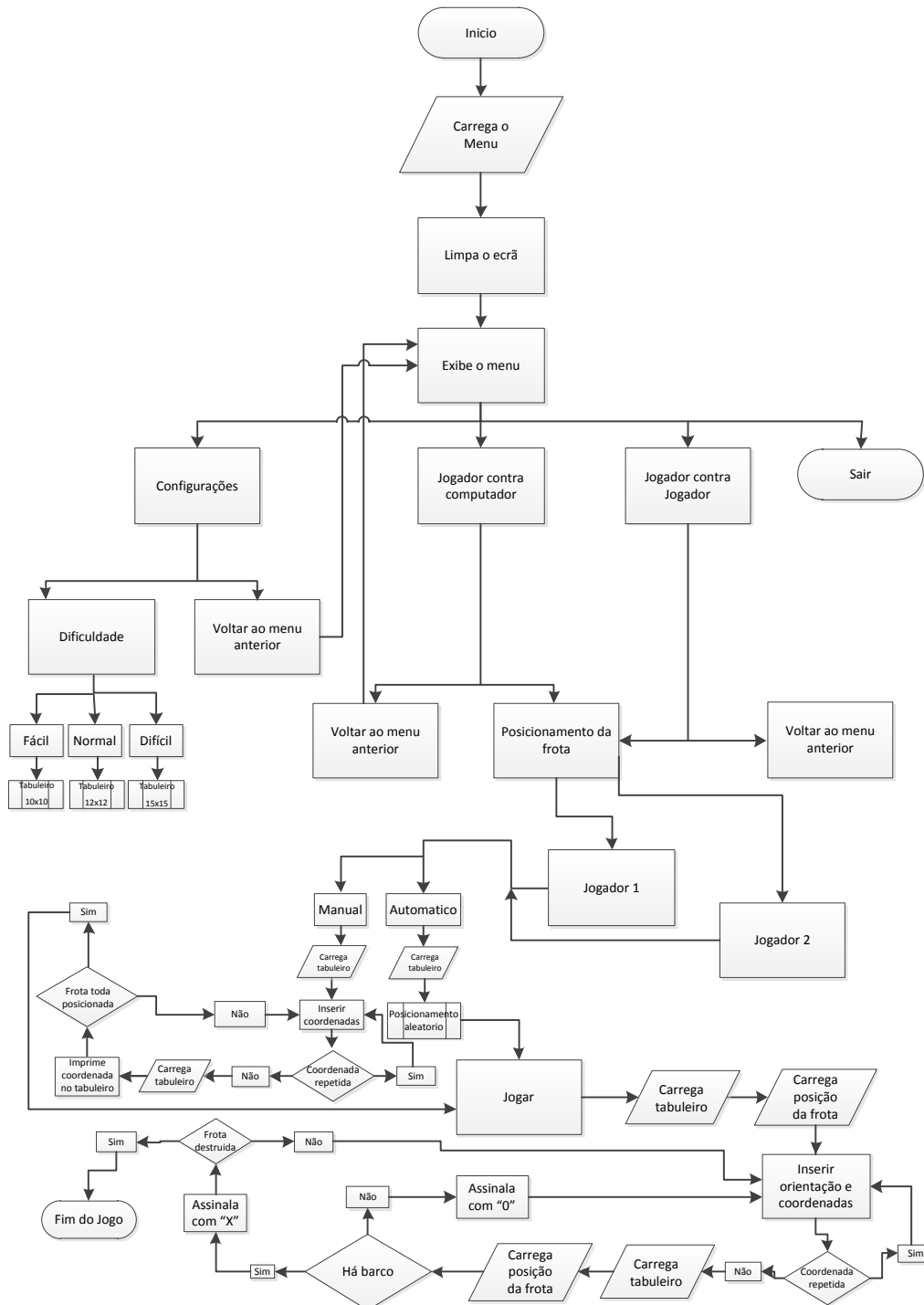


Figura 2 - Fluxograma do programa

2.2 - Definição de tarefas e atividades

A definição de tarefas aqui apresentada engloba todo o trabalho desenvolvido desde o início do projeto.

Na seguinte tabela estão descritas todas as tarefas que foram realizadas para a conclusão deste projeto consoante os requisitos e o fluxograma apresentado anteriormente.

Tabela 1 - Definição das tarefas

Código	Tarefas	
A	Gestão do projeto	
B	Definição dos objetivos	
C	Análise dos requisitos e elaboração do fluxograma	
D	Planificação do trabalho	
E	Desenvolvimento	
	1	Menus
	2	Representação visual do tabuleiro
	3	Posicionamento da frota
	4	Jogador contra computador
	5	Jogador contra jogador
	6	Fim do jogo
F	Integração das diferentes partes do código	
G	Testes	
H	Relatório	
I	Elaboração de uma apresentação do projeto	

A tarefa D foi dividida em sub-tarefas para que a divisão do código fosse possível. Cada elemento do grupo participou ativamente em uma ou mais componentes da aplicação.

2.3 - Divisão de tarefas

Com base nas tarefas definidas procedeu-se à divisão das mesmas, existindo tarefas comuns e individuais.

A tabela seguinte representa a divisão das tarefas:

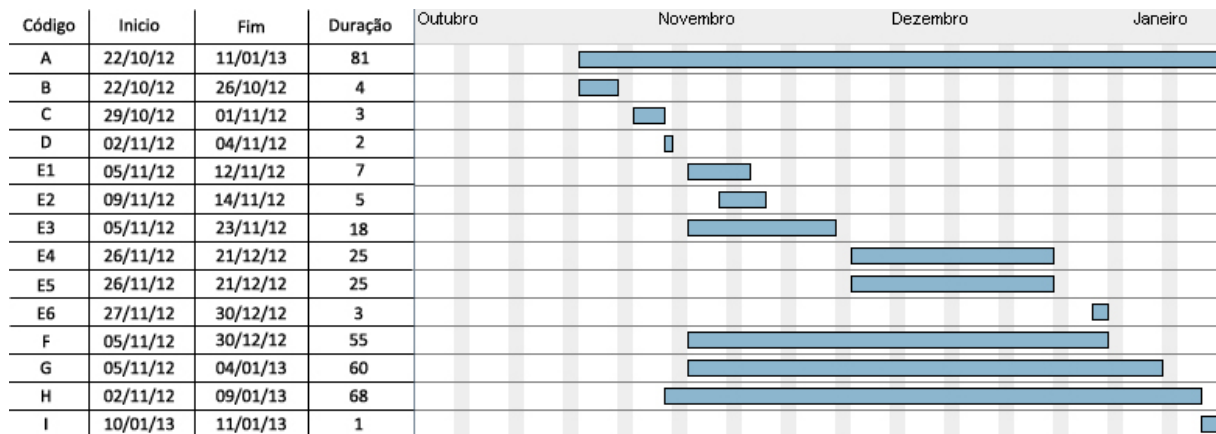
Tabela 2 - Divisão de tarefas

Código	Tarefas		Tempo previsto (dias)	Membro do grupo
A	Gestão do projeto		81	C, F, PA, PE
B	Definição dos objetivos		4	C, F, PA, PE
C	Análise dos requisitos e elaboração do fluxograma		3	C, F, PA, PE
D	Planificação do trabalho		2	C, F, PA, PE
E	Desenvolvimento		83	
	1	Menus		PE
	2	Representação visual do tabuleiro		PE
	3	Posicionamento da frota		F
	4	Jogador contra computador		C, F
	5	Jogador contra jogador		PA, PE
	6	Fim do jogo		PA
F	Integração das diferentes partes do código		55	PA,PE
G	Testes		60	C, F
H	Relatório		68	C, F,PA,PE
I	Elaboração de uma apresentação do projeto		1	C, F,PA,PE
C - Carlos Azinhaga F - Filipe Costa PA - Paulo Silva PE - Pedro Marquinhos				

2.4 - Calendarização de tarefas

A calendarização das tarefas está representada no seguinte diagrama de Gantt.

Tabela 3 - Calendarização das tarefas



2.5 - Recursos utilizados

Os recursos informáticos utilizados para o projeto foram os seguintes:

- Software Codeblocks - para compilação do código;
- Software MicrosoftVisio – para elaboração de fluxograma;
- Plataforma CodeUa - para consulta de informações, comunicação e repositório;
- Software GanttProject - elaboração de diagrama de Gantt;
- Plataforma Dropbox - para consulta de informações e repositório de versões (controle efetuado pelos elementos do grupo).

3- DESCRIÇÃO DAS ATIVIDADES DESENVOLVIDAS

Neste terceiro capítulo abordam-se três temas principais: a solução desenvolvida, os testes efetuados e a descrição do código. O primeiro tema foi dividido em dois subtemas: a interface gráfica e a descrição do programa

3.1 - Apresentação da solução desenvolvida

3.1.1 - Interface gráfica

Sendo o tema deste projeto "Jogo da Batalha Naval", em termos de interface gráfica é necessário um tabuleiro. O tabuleiro é constituído por letras na horizontal e números na vertical. Para as letras o grupo decidiu utilizar caracteres alfanuméricos; para os números, simplesmente números. Vejamos a seguinte tabela:

Tabela 4 - Simbologia utilizada para o tabuleiro

Descrição	Símbolo	Código ASCII
Letras	A	65
	B	66

	J	74

	L	76

	O	79
Números	1	
	2	
	...	
	10	
	...	
	12	
	...	
	15	

Note-se que o tamanho do tabuleiro pode variar consoante a dificuldade, daí as letras variarem entre "A" e "O" e os números entre 1 e 15.

Para simbolizar a frota no tabuleiro optou-se por uma representação de caracteres. O número de caracteres utilizados é equivalente ao número de posições que o barco ocupa no tabuleiro como demonstra a tabela seguinte.

Tabela 5 - Simbologia utilizada para a frota

Frota	Quantidade	Símbolo
Porta-aviões	1	PPPPP
Submarino	2	SSSS
Destroyer	3	DDD
Bote	4	B

Para distinguir se um tiro dado por um jogador acertou nalgum barco da frota ou na água utilizou-se a seguinte simbologia:

Tabela 6 - Simbologia utilizada para deteção de impacto

Tiro	Símbolo
Barco	X
Água	0

3.1.2 - Descrição do programa

Nesta secção será descrito o programa desenvolvido, explicando as suas funcionalidades. Mais à frente será apresentada detalhadamente a estrutura do código.

→ Início do jogo

O jogo tem início com a apresentação de um cabeçalho indicando o nome do jogo, “Batalha Naval”, e um menu com quatro opções de escolha par o utilizador, sendo elas:

- Configurações;
- Jogador contra computador;
- Jogador contra jogador;
- Sair.

→ Configuração do jogo

Seguidamente o utilizador poderá configurar o jogo. Para tal, será mostrado no ecrã duas opções

- Dificuldade (Fácil - F; Normal - N; Difícil - D);
- Voltar ao menu anterior.

Na opção “Dificuldade”, o programa irá criar uma tabuleiro de tamanho correspondente à dificuldade selecionada, sendo que “fácil” corresponde a uma tabuleiro de dimensão 10x10, “normal” a um tabuleiro de dimensão 12x12 e “difícil” a uma tabuleiro de dimensão 15x15.

É efetuada validação de dados. ✓

→ Jogador contra computador

Neste submenu do jogo o utilizador tem duas opções:

- Posicionamento da Frota (Automático - A; Manual - M);
- Voltar ao menu anterior.

Ao selecionar o posicionamento de frota, o jogador poderá escolher se quer que o posicionamento seja feito pelo computador, posicionamento automática, ou que seja o próprio a posicionar a sua frota, posicionamento manual.

No posicionamento manual é solicitado ao utilizador que escolha o sentido de posicionamento do barco, vertical (v) ou horizontal (h) e de seguida que introduza a coordenada alfanumérica da posição onde quer que o barco seja colocado.

A imagem seguinte mostra a interface de posicionamento manual da frota no nível de dificuldade “normal”:

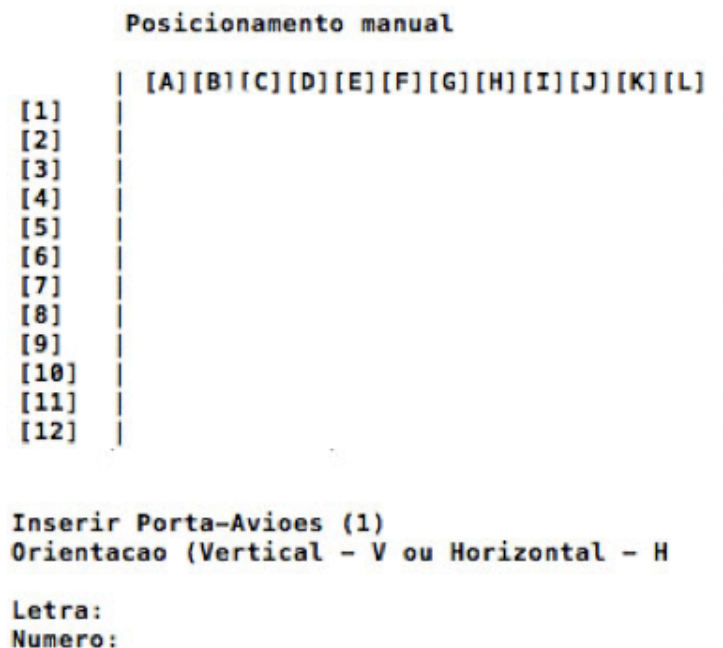


Figura 4 – Posicionamento manual

No final de ser efetuado o posicionamento, seja ele manual ou automático, é dado o início ao jogo.

O utilizador é o primeiro a jogar. É pedido ao utilizador que coloque a letra e número (figura 5) onde quer efetuar o disparo. No tabuleiro aparecerá os símbolos referentes à situação, se acertou num barco aparecerá um “X”, se acertou na água aparecerá um “0” (figura 6). Caso o utilizador repita a coordenada (letra e número) é-lhe dada a oportunidade de jogar outra vez, acrescentando uma jogada ao total de jogadas.

	[A]	[B]	[C]	[D]	[E]	[F]	[G]	[H]	[I]	[J]	[K]	[L]
[1]												
[2]												
[3]												
[4]												
[5]												
[6]												
[7]												
[8]												
[9]												
[10]												
[11]												
[12]												

Letra:
Numero:

Figura 5 – tabuleiro de jogo

	[A]	[B]	[C]	[D]	[E]	[F]	[G]	[H]	[I]	[J]	[K]	[L]
[1]												
[2]		0									X	
[3]												
[4]												
[5]												
[6]												
[7]												
[8]												
[9]												
[10]												
[11]												
[12]												

Letra:
Numero:

Figura 6 – tabuleiro com duas jogadas

As jogadas serão efetuadas alternadamente entre o utilizador e o computador, independentemente se o disparo acerta ou não no barco. Quando um dos jogadores destruir a frota adversária o jogo termina aparecendo no ecrã as mensagens “Parabéns! O Jogador X ganhou!”, “Destruiu toda a frota inimiga” e o número total de jogadas.

– Jogador contra jogador

Esta opção permite que um jogador humano jogue contra outro humano. Ao entrar neste menu, caberá ao Jogador 1, definido previamente pelos jogadores, escolher a dificuldade do jogo. Posteriormente o Jogador 1 irá escolher como quer colocar a sua frota, manual ou automaticamente. Após a colocação da frota do Jogador 1, será a vez do Jogador 2 posicionar a frota, contando também com as mesmas opções do Jogador 1, posicionamento manual (figura 3) ou automático. Após a colocação da frota pelo Jogador 2, é dado o início ao jogo.

É o Jogador 1 que inicia o jogo, e as jogadas serão efetuadas alternadamente, independentemente dos jogadores acertarem ou não num barco. No tabuleiro aparecerá os símbolos referentes à situação, se acertou num barco aparecerá um “X”, se acertou na água aparecerá um “0” (figura 5). Caso o utilizador repita a coordenada (letra e número) é-lhe dada a oportunidade de jogar outra vez, acrescentando uma jogada ao total de jogadas. O jogo termina quando um dos jogadores tiver destruído a frota adversária, aparecendo no ecrã a mensagem “Parabéns! O Jogador X ganhou!”, “Destruiu toda a frota inimiga” e o número total de jogadas..

3.2 - Descrição do código

Nesta secção será feita uma descrição detalhada do código desenvolvido. Foram agrupadas algumas funções devido há existência de algumas semelhanças entre as mesmas.

→ Funções utilizadas

Int main()

- Esta função disponibiliza o menu principal e estabelece ao tabuleiro uma dimensão padrão. O valor padrão da dimensão é passado por referencia, isto é, passa o endereço de memória do valor.

void menu(char tabuleiro[15][15][2], char frota[15][15][2], int *dim)

- Esta função vai imprimir 4 menus :
 1. Configuração;

2. Jogador contra computador;
3. Jogador contra jogador;
4. Sair.

void menu_subopcao1(char tabuleiro[15][15][2], char frota[15][15][2], int dim)

- Esta função vai imprimir as opções de configuração, nomeadamente a dificuldade do jogo, isto é, a dimensão (*dim*) do tabuleiro e a opção de voltar ao menu anterior.
 - Níveis de dificuldade:
 - *dim*=10 – fácil;
 - *dim*=12 – normal;
 - *dim*=15 – difícil.

void menu_subopcao2(char tabuleiro[15][15][2], char frota[15][15][2], int dim, int p, int nj)

- Função que permite ao jogador jogar contra o computador, disponibilizando duas opções posicionamento da frota (manual ou automático) e voltar ao menu anterior. Caso o posicionamento seja manual, será solicitado ao utilizador apenas a orientação (vertical ou horizontal) e a primeira coordenada do barco (figura 3), no caso automático a frota é gerada através da função *void desenha_barcos()*, explicada detalhadamente mais à frente.
- Gera automaticamente a frota do computador.

void menu_subopcao3(char tabuleiro[15][15][2], char frota[15][15][2], int dim)

- Função utilizada para jogar entre utilizadores. Utiliza a mesma metodologia do *void menu_subopcao2()*, mas pensado para jogo entre humanos.
- Disponibiliza duas opções:
 - posicionamento da frota;
 - voltar ao menu anterior.

voidl impar_matriz (char aux[15][15][2], int dim, int nj)

- Substitui todos dados gerados automaticamente ao criar a matriz por espaços em branco.

void desenha_matriz (char aux[15][15][2], int dim, int nj)

- Função que representa visualmente o tabuleiro de jogo na consola.

void limpa_ecra()

- Opção criada para evitar a constante utilização do nativa da Linguagem de programação C.

void desenha_barcos(char frota[15][15][2], int dim, int p, int orientacao, int tamanho, char character, int nj)

- Função que recebe como parâmetros a orientação do barco, o tamanho do mesmo e a letra que o identifica.
- São efetuadas validações para que os barcos não se sobreponham e não sejam posicionados fora do tabuleiro de jogo.

int gera_orientacao()

- Define a orientação aleatoriamente entre 0 e 1, sendo que 0 é horizontal e 1 é vertical.

int gera_coordenadas(int dim)

- Gera coordenadas aleatoriamente entre 0 e *dim* (dimensão do tabuleiro).

int verifica_orientacao()

- Converte um *char* para um *int*, ou seja, V (vertical) = 1 e H (horizontal) = 0.

void posi_frota(char frota[15][15][2], int dim, int p, int nj)

- Representação visual da frota consoante a orientação, tamanho do barco e a letra que o identifica.

void jogar(char tabuleiro[15][15][2], char frota[15][15][2], int dim, int p, int nj)

- Detecção do local de impacto, marcando com “X” se acertar num barco, caso contrario marca com “0”;
- Estatística do jogo numero de jogada numero de barcos abatidos

void jogar2(char tabuleiro[15][15][2], char frota[15][15][2], int dim, int p, int nj)

- Função idêntica à anterior, mas para o Jogador 2.

3.3 - Testes efetuados

Ao longo do projeto, consoante a implementação do código, foram efetuados testes sucessivos para verificar o correto funcionamento do programa. Assim verificávamos se estávamos a cumprir o requisitos mínimos.

4 - ANÁLISE DE RESULTADOS

→ Menus

Durante o percurso do projeto várias alterações foram efetuadas , sendo uma das mais importantes a passagem por referencia dos valores entre diferentes menus.

→ Representação visual do tabuleiro

A representação do jogo é constituída por Letras(Horizontal) e números (Vertical) não é composta por nenhuma livraria.

→ Número de tiros

O numero de tiros inicialmente foi definido como 1 ou 3 tiros, consoante a escolha do utilizador. Contudo, com alguns problemas encontrados do decorrer do projeto, foi descartada a opção de 3 tiros, isto porque ao serem efectuados os três tiros foram encontrados vários erros nas funcionalidades no tipo de jogo “Jogador contra Computador”.

→ Jogador contra Computador

Visto que é uma das partes com mais complexidade do projeto, optámos por implementar este ponto em primeiro lugar e seguidamente aproveitar a sua estrutura para a implementação da funcionalidade “Jogador contra Jogador”.

Cabe ao computador construir o seu tabuleiro, que é composto por uma frota disponibilizada automaticamente.

Os tiros efectuados pelo computador são gerados de forma aleatória, em que não são permitidas as mesmas coordenadas anteriormente utilizadas.

→ Jogador contra jogador

As definições do jogo são pedidas ao Jogador 1, ou seja, é este jogador que de entre as opções disponíveis, fácil (10x10), normal (12x12) e difícil (15x15), escolhe a dificuldade do jogo. É permitido aos utilizadores posicionar a sua frota automática ou manualmente, em que manualmente é pedido ao utilizador para escolher a orientação, vertical (V) ou horizontal (H) do seu barco, sendo apenas pedido a primeira coordenada (letras e números) (figura 3) e depois cabe ao programa completar as restantes posições do barco (figura 4).

→ **Fim do jogo**

É composto pelas estatísticas do jogo, isto é, o jogador vencedor vê quantas jogadas foram efectuadas pelos utilizadores e, o numero de barcos abatidos.

5 – CONCLUSÃO

Tabela 7–Requisitos concluídos

Requisitos	Concluído
Representação visual do tabuleiro	✓
Dois jogadores humanos	X
Jogador contra computador	X
Deteção do local do impacto	✓
Fim do jogo	✓
Configuração do tamanho do tabuleiro	X

Com base na tabela acima, será efectuada uma reflexão critica do trabalho desenvolvido, dos requisitos propostos, das soluções encontradas e dos principais problemas.

Dos quatro requisitos inicialmente propostos, três foram concluídos, nomeadamente a representação visual do tabuleiro e a detecção de impacto e fim do jogo, não tendo sido concluído o requisito de jogo entre dois jogadores humanos.

Para além dos requisitos mínimos referidos no enunciado do projeto, foram estipulados por nós mais dois requisitos: jogador conta computador e configuração do tamanho do tabuleiro. Destes dois, nenhum deles foi concluído com sucesso.

A fase de testes, efectuada ao longo de todo o desenvolvimento do projeto, permitiu-nos encontrar erros. A tarefa de integração das diferentes partes do código permitiu-nos a sua optimização evitando assim repetição do mesmo.

Um dos aspectos que poderia ser melhorado no futuro seriam as validações dos parâmetros de entrada introduzidos pelo jogador e ainda da colocação da frota no tabuleiro (ex. uma casa de separação entre cada barco).

Uma ambição futura seria a conclusão das tarefas que ficaram por concluir.

6 – BIBLIOGRAFIA

LIVROS

DAMAS, Luís – **Linguagem C** 20ª edição FCA – Editora informática, 1999. ISBN 9789727221561.

TUTORIAIS

SÁ, Marcela Rocha Tortureli de (2012, 04 Dezembro). Apostila de Introdução à Linguagem C [Consult.]:<http://www.ufjf.br/petcivil/files/2009/02/Apostila-de-Introdução-à-Linguagem-C.pdf>

MARTINS, Luiz Gustavo Almeida (2012, 02 de novembro). Apostila de Linguagem C (Conceitos Básicos) [Consult.] http://www.facom.ufu.br/~gustavo/ED1/Apostila_Linguagem_C.pdf

SITE WEB

VITÓRIA, Deise.(2012, 24 Outubro) Blog Deise Vitoria [Em Linha]. Disponível em:
<http://deise.info/?p=1006>