**Этап 1. Анализ**

**1.1. Описание предметной области**

Морской бой – это классическая настольная игра, в которой два игрока соревнуются друг с другом, пытаясь потопить корабли противника на специально размеченном поле. Игра имеет долгую историю и широкую популярность во всем мире.

Начнём с первого: морской бой – довольно молодая игра. Версии истории её появления разнятся, однако можно выделить средний временной промежуток появления её на свет – конец 19 – начало 20 века. По одной из версий игра была изобретена в 1970-х годах бурлаком Петром Кондратьевым, как способ отвлечься от тяжёлого труда (подробно рассмотрено в [1, 2]). С тех пор она получила широкое распространение, эволюционировала и успешно дошла до наших дней.

Обычные правила морского боя просты и понятны даже для новичков. Игровое поле обычно представляет собой сетку, где вертикальные столбцы обозначают буквы, а горизонтальные строки – цифры. К примеру, в одной из простейших вариации игры всё это расчерчивается на клеточной бумаге. Затем каждый игрок размещает одинаковый набор кораблей на своей половине поля, где суда не видны противнику. Корабли имеют разную длину и форму, от одноклеточных до нескольких клеток в длину (и даже до нескольких в ширину – в зависимости от правил конкретной версии игры). Цель игры – отгадать координаты кораблей противника и потопить их все, стреляя в выбранные клетки поля.

Классическая версия морского боя, как правило, состоит из двух игровых полей размера 10x10 клеток, обозначенных в виде сетки. Каждый игрок получает своё поле и размещает на нём свои корабли перед началом игры.

В классической версии используются следующие типы кораблей с заданными размерами: один корабль из четырех клеток (большой), два корабля из трех клеток (средние), три корабля из двух клеток (малые) и четыре корабля из одной клетки (однопалубные). Корабли могут быть размещены вертикально или горизонтально, но они не могут соприкасаться друг с другом ни углами, ни сторонами.

Геймплей довольно прост: игроки по очереди называют координаты на поле противника, пытаясь попасть в корабли противника. Например, игрок может сказать "В-3" для атаки клетки B3 на поле противника. Если выстрел попадает в пустую клетку, то игрок сообщает "Мимо". Если выстрел попадает в корабль противника, игрок сообщает "Попал". Противник затем сообщает, был ли корабль потоплен, с помощью слова "Потоплен". Если атака была успешной (игрок попал по кораблю противника), то игрок снова делает ход, игнорируя очерёдность. Игроки отмечают результаты выстрелов на своих полях, чтобы отслеживать уже сделанные ходы. Побеждает тот игрок, который сумеет первым потопить все корабли противника.

В современных версиях морского боя можно встретить различные вариации правил и дополнительные элементы. Например, некоторые версии добавляют возможность использования специальных умений или различных типов кораблей с разными характеристиками. Могут быть введены дополнительные элементы стратегии, такие как использование бонусов или специальных атак. Некоторые издания игры также предлагают разные тематические варианты, связанные с фильмами, книгами или компьютерными играми.

Морской бой также был успешно адаптирован для компьютерной игры. Такие версии оригинальной игры могут предлагать как одиночный геймплей против компьютерного противника, так и многопользовательский режим, где игроки могут соревноваться онлайн. Реализации морского боя также доступны на мобильных устройствах и позволяют играть в любое время и в любом месте. Рассмотрим эти версии поближе.

**1.2. Обзор аналогов**

Вся представленная далее информация актуальна только на момент презентации этой работы.

1. Название проекта: "Battleship"

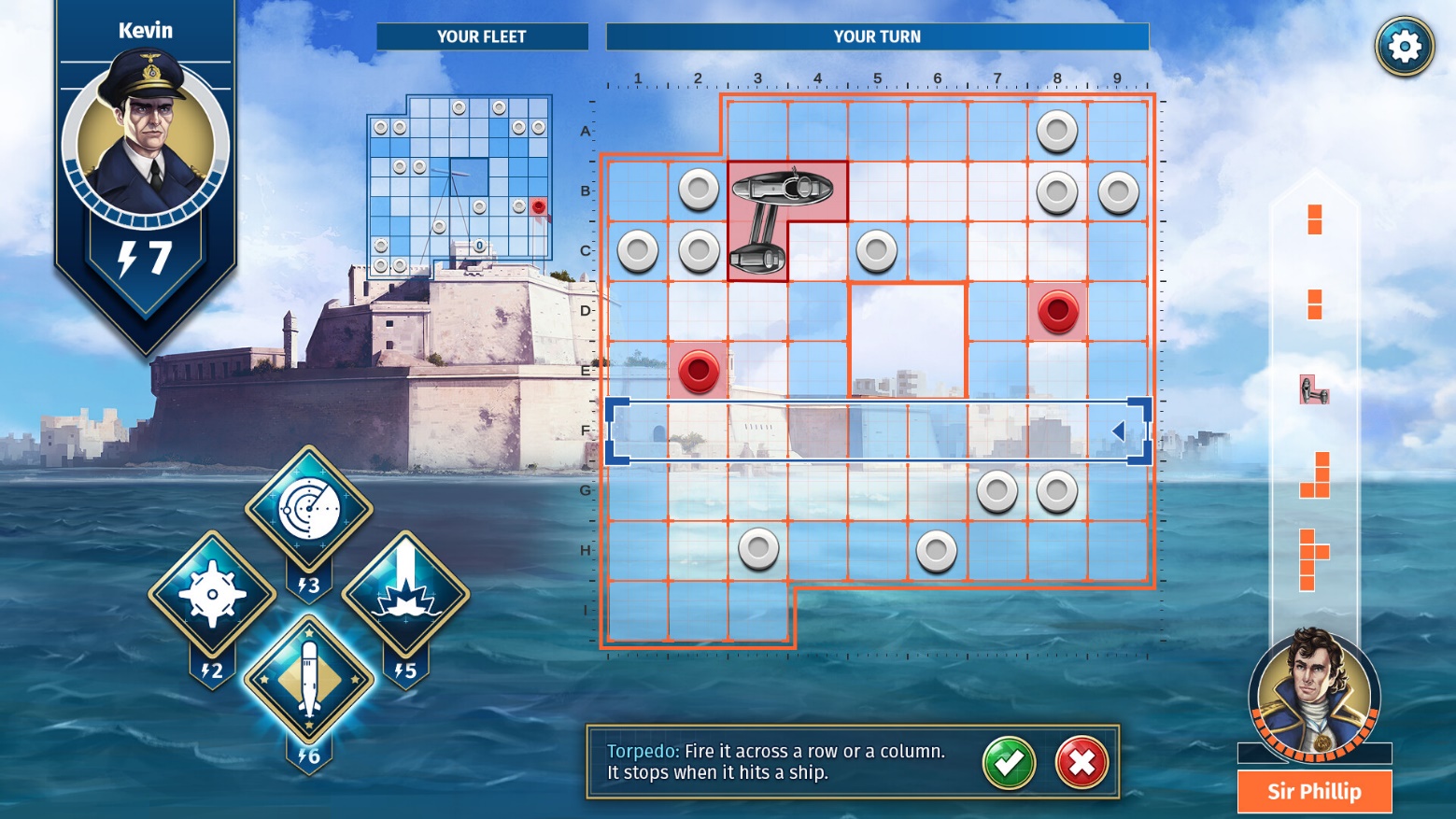


Рисунок 1. Геймплей первой игры-аналога.

* Разработчик проекта: Marmalade Game Studio
* Год выпуска: 2016
* Статус проекта: По-прежнему доступен
* Наличие онлайн-режима: Да
* Платформы: ПК (Windows), мобильные устройства (iOS, Android)
* Онлайн-сервисы: Steam.
* Отзывы Steam: в основном положительные.
* Плюсы:
  + Обновленная графика и улучшенный визуальный опыт.
  + Возможность игры с компьютерным противником и онлайн-мультиплеером.
  + Различные режимы и настройки для более интересного игрового опыта.
* Минусы:
  + Могут возникать некоторые проблемы с соединением при игре в онлайн-режиме.
  + Некоторые игроки могут посчитать, что игра не достаточно инновационна и предлагает стандартный опыт игры в "Морской Бой".
* Общее представление: "Battleship" от Marmalade Game Studio предлагает классическую игру "Морской Бой" с обновленной графикой и возможностью играть как с компьютерным противником, так и с реальными соперниками онлайн. Игра имеет различные режимы и настройки, чтобы сделать игровой опыт более интересным и разнообразным. В том числе имеет различные размеры и формы игрового поля.

1. Название проекта: "Sea Battle 2"

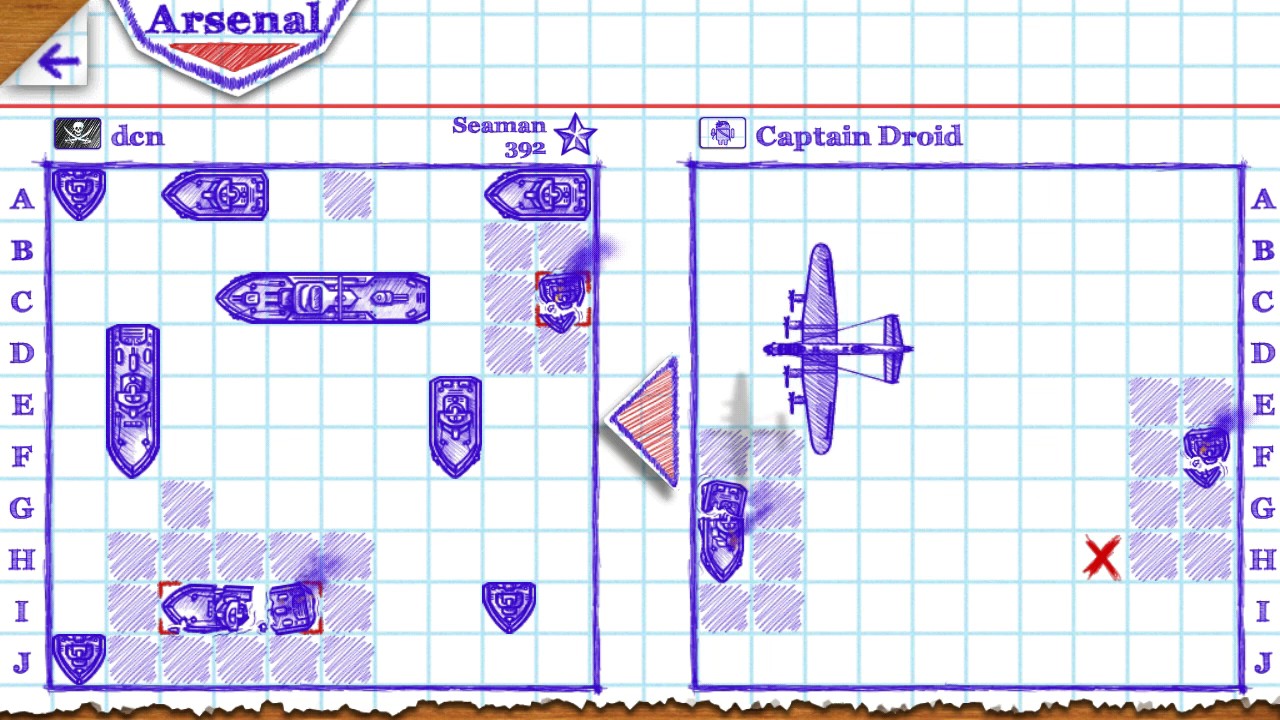


Рисунок 2. Геймплей второй игры-аналога

* Разработчик проекта: BYRIL
* Год выпуска: 2014
* Статус проекта: По-прежнему доступен
* Наличие онлайн-режима: Да
* Платформы: мобильные устройства (iOS, Android)
* Онлайн-сервисы: App Store, Google Play.
* Рейтинг: Google Play – 4.7/5; App Store – 4.7/5.
* Плюсы:
  + Улучшенная графика и множество различных кораблей и специальных орудий.
  + Возможность игры с компьютерным противником и мультиплеером онлайн.
  + Гибкие настройки и возможность создания собственных правил игры.
* Минусы:
  + Некоторые игроки могут считать, что геймплей становится монотонным со временем.
  + Иногда могут возникать проблемы с соединением или задержкой в онлайн-режиме.
* Общее представление: "Sea Battle 2" от BYRIL предлагает современную интерпретацию классической игры "Морской Бой". Она имеет улучшенную графику, множество различных кораблей и специальных орудий. Игра позволяет играть с компьютерным противником или против реальных игроков в онлайн-режиме. Также есть возможность создавать свои собственные правила и настраивать игровой процесс.

1. Название проекта: "Battleship"

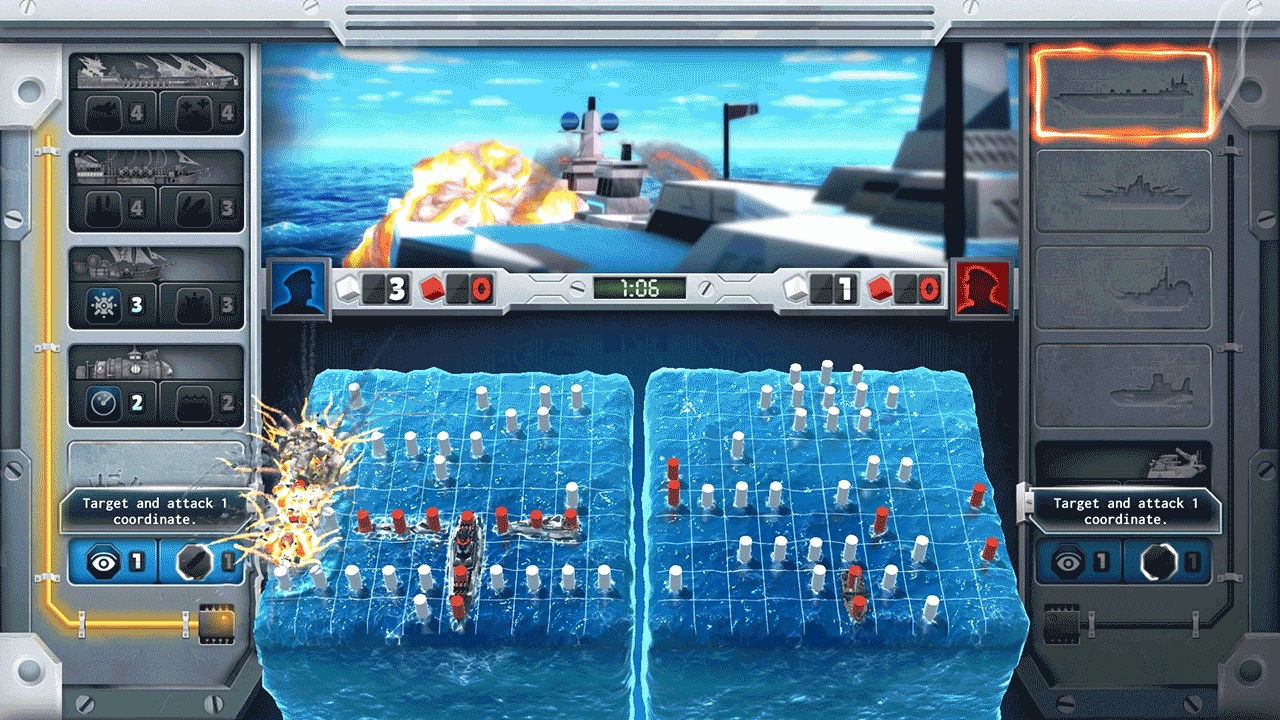


Рисунок 3. Геймплей третьей игры-аналога

* Разработчик проекта: Ubisoft
* Год выпуска: 2012
* Статус проекта: По-прежнему доступен
* Наличие онлайн-режима: Да
* Платформы: приставки (PlayStation 4, Xbox One, Wii)
* Онлайн-сервисы: PlayStation Network, Xbox Live, официальный онлайн-магазин Ubisoft
* Рейтинг: Разнообразный рейтинг в зависимости от платформы.
* Плюсы:
  + Привлекательная графика и качественное визуальное оформление.
  + Возможность игры с компьютерным противником и мультиплеером онлайн.
  + Режим кампании с сюжетной линией, связанной с фильмом "Battleship".
* Минусы:
  + Игра может быть слишком простой и не предлагать достаточно сложных тактических решений.
  + Некоторые игроки могут считать, что игра ограничена в разнообразии и долгоиграемости.
* Общее представление: "Battleship" от Ubisoft – это адаптация игры "Морской Бой" в виде видеоигры. Игроки могут играть как с компьютерным противником, так и против других игроков в многопользовательском режиме онлайн. Игра предлагает различные режимы и настройки, а также имеет режим кампании с сюжетной линией, связанной с фильмом "Battleship".

**1.3. Обобщение частной задачи**

Написать прототип игры “Морской бой” для командной строки с возможностью добавления сетевых функций, таких как сетевая игра и таблица рекордов.

Форма поля и его размеры планируются быть константными. Само поле должно быть представлено массивом символов. Количество кораблей, их размеры и форма также планируются быть константными. Набор кораблей должен быть представлен массивом структур.

Задачи:

* Реализовать многопользовательский геймплей (формат PvP – человек против человека)
* Рассмотреть возможность добавления сетевых функций, а также реализовать их в случае, если такая возможность есть.
* Также в случае добавления сетевых функций добавить авторизацию пользователей, а также таблицу рекордов
* Реализацию нужно выполнить под Windows без использования фреймворков
* Реализовать игру, провести QA-тестирование

**Этап 2. Проектирование**

В процессе проектирования выяснилось, что создание клиент-серверной архитектуры с сетевыми возможностями было довольно сложной и комплексной задачей, для решения которой были необходимы более обширные знания и умения, поэтому от этой идеи было решено отказаться.

**2.1. Формальное описание задачи**

В дополнение к обобщению частной задачи (Этап 1.2) можно выделить, что правила будут соответствовать классическому морскому бою:

* Размер поля 10 на 10 клеток
* 1 большой корабль (1 на 4 клетки)
* 2 средних корабля (1 на 3 клетки)
* 3 малых корабля (1 на 2 клетки)
* 4 однопалубника (1 на 1 клетку)

**2.2. Поведенческая модель программы**

Взаимодействие с пользователем предполагается производить через текстовое меню, открывающееся в командной строке при запуске программы. Игрокам нужно предоставить основную информацию о программе, игре, а также необходимо ознакомить их с правилами.

В начале игры расстановка кораблей игроками должна проводится по очереди –сначала первый игрок полностью расставляет корабли на своём поле, а затем второй – следующим образом:

1. Игрок вводит координату носа первого корабля (например: «E1»)
2. Игрок вводит направление расположения остальной части корабля относительно его носа (например: «down»)

При этом программа должна воспринимать как заглавные, так и строчные буквы при вводе.

После расстановки всех кораблей каждым игроком должен начинаться основной игровой цикл. Определение игрока, который будет проводить атаку первым необходимо сделать случайным, чтобы исключить постоянное преимущество у какого-либо пользователя.

Атака игроками будет проводится аналогично расстановке кораблей – пользователь вводит координату клетки для атаки (например: «D6»). О попадании в корабль или промахе мимо него будет сообщаться, а также будет сообщаться о потоплении корабля или победе игрока.

**2.3. Представление данных в памяти при их обработке**

Каждое поле должно быть представлено в виде двойного массива символьного типа 10 на 10. Координаты клетки для атаки или для установки носа корабля для передачи между функциями программы будут представлены в виде одной структуры, в которой будет две целочисленных переменных – координаты клетки по осям. Также набор кораблей будет представлен в виде массива структур, где каждый элемент массива будет представлять данные по одному из типов кораблей. Сама же структура должна иметь следующие данные: размер судна (целочисленный тип), количество судов этого типа (целочисленный тип) и его название (символьный тип, строка).

* 1. **Основные алгоритмы**

Рассмотрим основные алгоритмы, которые будут представлены в программе:



Схема 1. Алгоритм работы основного меню

Основное контекстное меню будет предлагать пользователю ознакомиться с правилами, информацией о разработчике, начать игру или же выйти из неё. Цикл while с условием, которое всегда истинно необходим для возможности возврата в главное меню из подпунктов меню, а также из основной игровой функции.



Схема 2. Общий алгоритм работы игровой функции

Здесь стоит отметить, что однотипные действия, по типу инициализации полей разбиты на отдельные функции в соответствии с общими принципами программирования KISS и DRY (подробно рассмотрено в [3]).

Основной цикл while со всегда верным условием используется для возможности начать игру заново без использования нерекомендуемых конструкций (по типу goto), а вложенный для повторения ходов до победы одного из игроков.

Возвращаемое функцией атаки значение в случае победы одного из игроков будет соответствовать номеру победившего игрока, в обратном же случае оно должно быть равно нулю. Также подразумевается, что игрок, который успешно провёл атаку, то есть попал по кораблю противника, должен снова делать ход (то есть атакующий не меняется) в случае удачной атаки (попадания по кораблю противника) игрок должен снова делать ход, без смены атакующего (описано в п.1.1). Поэтому возвращаемое значение функции атаки для такого случая должно быть отлично от нуля и отлично от номеров игроков (в стандартном случае 1 и 2), оно может быть равно, например, трём. В последнем условии if для переменной, значение которой равно возвращённому функцией как раз проверяется такой случай.



Схема 3. (потом уедет в приложения) Общий алгоритм работы функции атаки

Здесь под «клетка пуста» понимается отсутствие части корабля в ней. Также, потопленные части кораблей и промахи на поле будет обозначаться разными символами для того, чтобы:

1. Наглядно выводить поле для пользователя
2. Для правильной работы алгоритма распознавания потопленного судна, а также для правильного обозначения клеток вокруг потопленных судов как атакованных.

Также здесь клетки вокруг потопленных кораблей будут отмечаться как атакованные (но с промахом) с целью более понятного обозначения потопленных кораблей на поле и упрощения геймплея для игроков.

Для оптимизации проверка на конец игры будет проводиться только в случаях потопления судна, так как только при потоплении всех судов игра будет считаться завершённой, то есть вывод о завершении игры можно сделать только после уничтожения какого-то корабля.

**Этап 3. Разработка программного продукта**

Ниже приведены листинги программы

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <stdbool.h>

#include <string.h>

Листинг 1. Используемые библиотеки

#define FIELD\_SIZE 10

#define NUM\_SHIPS 4

Листинг 2. Используемые макросы

Макросы используются для упрощения работы с однотипными данными (размеры поля, количество типов судов).

typedef char Field[FIELD\_SIZE][FIELD\_SIZE];

11:

typedef struct {

int x;

int y;

} Coord;

typedef struct {

int size;

int num\_ships;

char\* name;

} Ship;

Листинг 3. Объявления новых имён для типов и групп типов данных (двух структур)

1. void print\_field(Field field, bool show\_ships);
2. void init\_field(Field field);
3. int atack(Field defender\_field, Ship\* ships, int current\_player);
4. bool place\_ship(Field field, Ship\* ship, Coord coord, char direction);
5. bool is\_valid\_position(Field field, Ship\* ship, Coord coord, char direction);
6. Ship\* get\_ship\_by\_size(int size, Ship\* ships);
7. void print\_ships(Ship\* ships);
8. void init\_ships(Ship\* ships);
9. Coord get\_coord\_input(char\* prompt);
10. char get\_direction\_input(char\* prompt);
11. bool is\_coord\_on\_field(Coord coord);
12. Ship\* is\_ship\_sunk(Field field, Coord coord, Ship\* ships, int\* cells\_zero, int N);
13. bool is\_game\_over(Field field);
14. void main\_menu();
15. void game();
16. void about\_game();
17. void about\_dev();
18. void place\_ships(Field field, Ship\* ships, int player);

Листинг 4. Объявление всех функций, использующихся в программе

Функции объявлены в начале, так как описание всех функций для удобства сделано после описания главной функции (main)

43: int main() {

44: system("chcp 1251 > null");

45: main\_menu();

46: return 0;

47: }

Листинг 5. Описание главной функции

49: void place\_ships(Field field, Ship\* ships, int player)

50: {

51: Coord coord;

52: int i = 0;

53: char direction = 0, c = 0;

54: for (i = 0; i < NUM\_SHIPS; i++) {

55: Ship\* ship = &ships[i];

56: direction = 0;

57: for (; ship->num\_ships > 0; ship->num\_ships--) {

58: printf("Player %d, place your ships:\n", player);

59: printf("Available ships:\n");

60: print\_ships(ships);

61:

62: printf("Field:\n");

63: print\_field(field, true);

64:

65: printf("Place your %s ship (amount: %d)\n", ship->name, ship->num\_ships);

66:

67: coord = get\_coord\_input("Enter ship's nose coordinate (e.g. A1): ");

68: if (ship->size != 1)

69: direction = get\_direction\_input("Enter ship's direction (up, down, 70: left or right): ");

71: if (!place\_ship(field, ship, coord, direction)) {

72: ship->num\_ships++;

73: printf("Press [Enter] to continue\n");

74: while ((c = getchar()) != '\n' && c != EOF);

75: }

76: system("cls");

77: }

78: }

79: }

Листинг 6. Описание функции расстановки всех кораблей на поле игрока

void about\_game()

{

char c = 0;

printf("Морской бой – это классическая настольная игра, в которой два игрока соревнуются друг с другом,\n"

"пытаясь потопить корабли противника на специально размеченном поле.\n"

"Игровое поле представляет собой сетку, где вертикальные столбцы обозначают буквы, а горизонтальные строки – цифры.\n\n"

"В начале игры каждый игрок размещает корабли на своём поле, где они не видны противнику.\n"

"Игроки по очереди должны атаковать клетки на поле противника, при попадании будет выводится 'Hit!', при промахе - 'Miss!'.\n"

"Если корабль противника потоплен, будет выведено соответствующее сообщение об уничтожении судна с его названием.\n"

"Побеждает тот, кто первым потопит все корабли противника.\n\n"

"Нажмите клавишу [Enter] для продолжения:\n");

while ((c = getchar()) != '\n');

return;

}

* + 1. Название: История морского боя

Автор: Неизвестен

Интернет-адрес: <https://istorygames.info/logicheskie-igryi/istoriya-morskogo-boya.html>

* + 1. Название: Морской бой

Автор: Неизвестен

Из журнала «Игры и Игрушки» №5-2017

Интернет-адрес: <https://www.i-igrushki.ru/archive/morskoy-boy5-17.html>

* + 1. Название: 7 Common Programming Principles That Every Developer Must Follow

Автор: anuupadhyay

Интернет-адрес: <https://www.geeksforgeeks.org/7-common-programming-principles-that-every-developer-must-follow/>