**М** Государственное образовательное учреждение высшего профессионального образования

Московский государственный технический университет

им. Н.Э. Баумана

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

КУРСОВОЙ ПРОЕКТ

Искусственный интеллект для консольной игры

«Морской бой»

Руководитель проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Бородин А.А.

(подпись, дата)

Разработчик проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Забродина М.П.

(подпись, дата)

г. Москва 2020 г.

Оглавление

[Цель 3](#_Toc59812533)

[Введение 3](#_Toc59812534)

[Требования к проекту 3](#_Toc59812535)

[Проектирование игры 4](#_Toc59812536)

[Работа программы 7](#_Toc59812537)

[Начало игры 7](#_Toc59812538)

[Промежуточные этапы игры 9](#_Toc59812539)

[Конец игры 12](#_Toc59812540)

[ЗАКЛЮЧЕНИЕ 12](#_Toc59812541)

# Цель

Создание консольной игры «Морской бой» на алгоритмическом языке программирования С++ при помощи знаний, полученных на 1 курсе по специальности «Информационная безопасность». Разработка искусственного интеллекта.

# Введение

“Морской бой” -игра для двух участников, в которой игроки по очереди называют координаты на неизвестной им карте соперника. Если у соперника по этим координатам имеется корабль (координаты заняты), то корабль или его часть «топится», а попавший получает право сделать ещё один ход.

Цель игрока — первым потопить все корабли противника.

# Требования к проекту

Данная игра должна удовлетворять следующим требованиям:

* Соблюдение правил игры “Морской бой”:
  + количество кораблей должно строго соответствовать правилам;
  + корабли на поле не должны касаться друг друга даже углами;
  + соблюдены правила, касающиеся размеров поля
  + правильная система координат
  + правильный ввод координат(сначала буква, потом цифра)
  + нельзя править расстановку кораблей после начала игры
  + пропуск хода недопустим
  + при попадании в корабль игроком, предоставляется право сделать ещё один ход
* Расчёт ходов искусственного интеллекта при всех возможных вариантах развития хода игры
* Автономность искусственного интеллекта, т. е. работа искусственного интеллекта без вмешательства человека

# Проектирование игры

Код данной консольной игры содержит класс “Поле”(Pole). “Поле” содержит следующие данные:

* vector<vector<int>> pole – поле с кораблями игрока
* int shots[10][10] – поле выстрелов игрока
  + Обозначения:
    - 0 – ячейка, по которой еще не стреляли
    - 1 – выстрел мимо корабля
    - 2 – ранение корабля
    - 3 – убитый корабль
    - 4 – рамка вокруг убитого корабля
* int AdrKor[5][10] – двумерный массив, содержащий данные о кораблях:
  + начальные координаты корабля
  + положение корабля на поле(горизонтальное или вертикальное)
    - 1 – вертикальное
    - 2 - горизонтальное
  + количество палуб
  + индикатор состояния корабля
    - 0 – корабль не тронут
    - 1 – корабль ранен
    - 2 – корабль убит

Функциями класса являются:

* Pole(vector<vector<int>> m){} – инициализация игрового поля на основе двумерного вектора, содержащего данные о ячейках поля
* void clear() – очистка поля с целью перестановки кораблей
* void print() – вывод поля с кораблями

Код так же содержит следующий ряд функций:

* int poiskAdresa(int row, int col, int pal, int polog, int user){}
  + функция проверяет, можно ли в указанные координаты(col, pal) с заданным количеством палуб(pal) и выбранным расположением корабля(polog) поставить корабль на поле игрока(user)
* void pointKor(int row, int col, int pal, int polog, int user){}
  + функция расставляет на поле корабль с координатами начала (row, col), количеством палуб(pal), расположением(polog) игрока(user)
* int read(char y){}
  + функция переводит из буквенного обозначения координаты в численное
* void pointKorablVruchnuy(){}
  + функция для расстановки кораблей на игровом поле игроком
* void pointKorablPC(int pal, int colum){}
  + функция для расстановки кораблей компьютера
* void pointHardPC(){}
  + функция для расстановки кораблей компьютера с повышенной сложностью игры
* void pointKorabl(int pal, int colum){}
  + функция для автоматической расстановки кораблей игрока(с возможностью повторной генерации, если игрока не устроила расстановка, заданная компьютером)
* int find\_korPC(int x, int y){}
  + функция для поиска раненого или убитого корабля компьютера при выстреле, изменения данных о кораблях и возвращаемым значением-“статусом” выстрела(мимо, ранен или убит)
* int find\_korIg(int x, int y){}
  + функция для поиска раненого или убитого корабля игрока при выстреле, изменения данных о кораблях и возвращаемым значением-“статусом” выстрела(мимо, ранен или убит)
* int Pobeda(){}
  + функция для проверки победы игрока и компьютера, возвращаемые значения: 0 – игра не закончена, 1 – выиграл игрок, 2 – выиграл компьютер
* void kill\_or\_not(Pole& m, Pole& n){}
  + функция для проверки индикатора состояния кораблей после выстрела
* void printpole(){}
  + функция вывода игрового поля на экран
* void HodIgroka(){}
  + функция хода игрока
* void HodPC(){}
  + функция хода компьютера

# Работа программы

## Начало игры

На рисунке 1 представлено начало работы программы



Рисунок 1

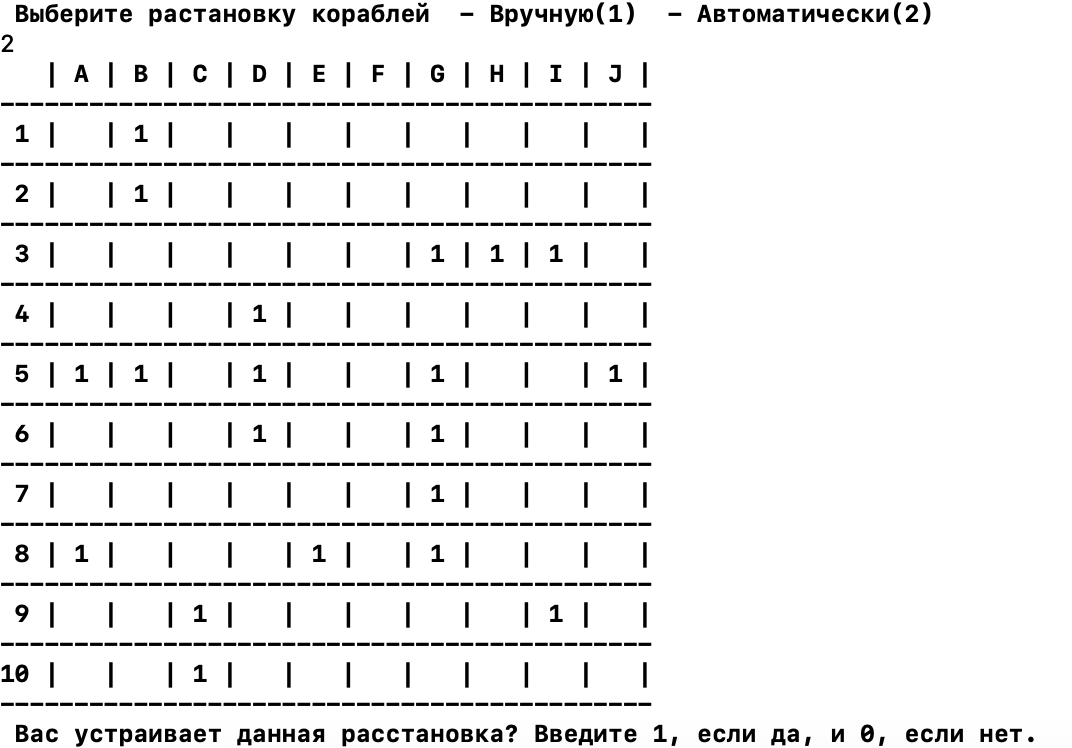


Рисунок 2

На рисунке 2 представлен пример автоматической расстановки кораблей игрока. Если игрока не устраивает данная расстановка, генерируется новая расстановка кораблей до тех пор, пока игрок не будет доволен.

На рисунке 3 представлен пример простой расстановки кораблей компьютера

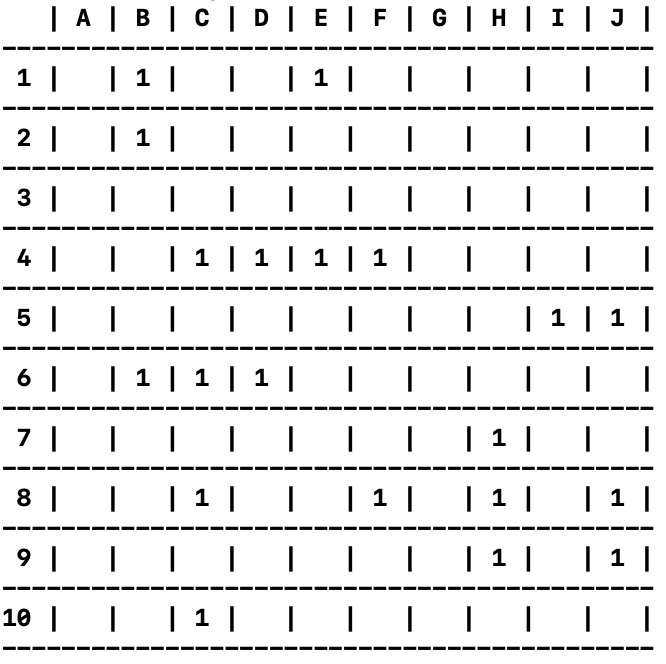


Рисунок 3

На рисунке 4 представлен пример более сложной расстановки кораблей компьютера



Рисунок 4

## Промежуточные этапы игры

В случае, если в результате действий игрока, корабль компьютера оказывается потопленным, вокруг него возводится рамка, обозначенная ‘.’. Это означает, что в помеченных ячейках корабли находиться не могут, иначе это перечило бы правилам игры. Данный случай представлен на рисунке 5.

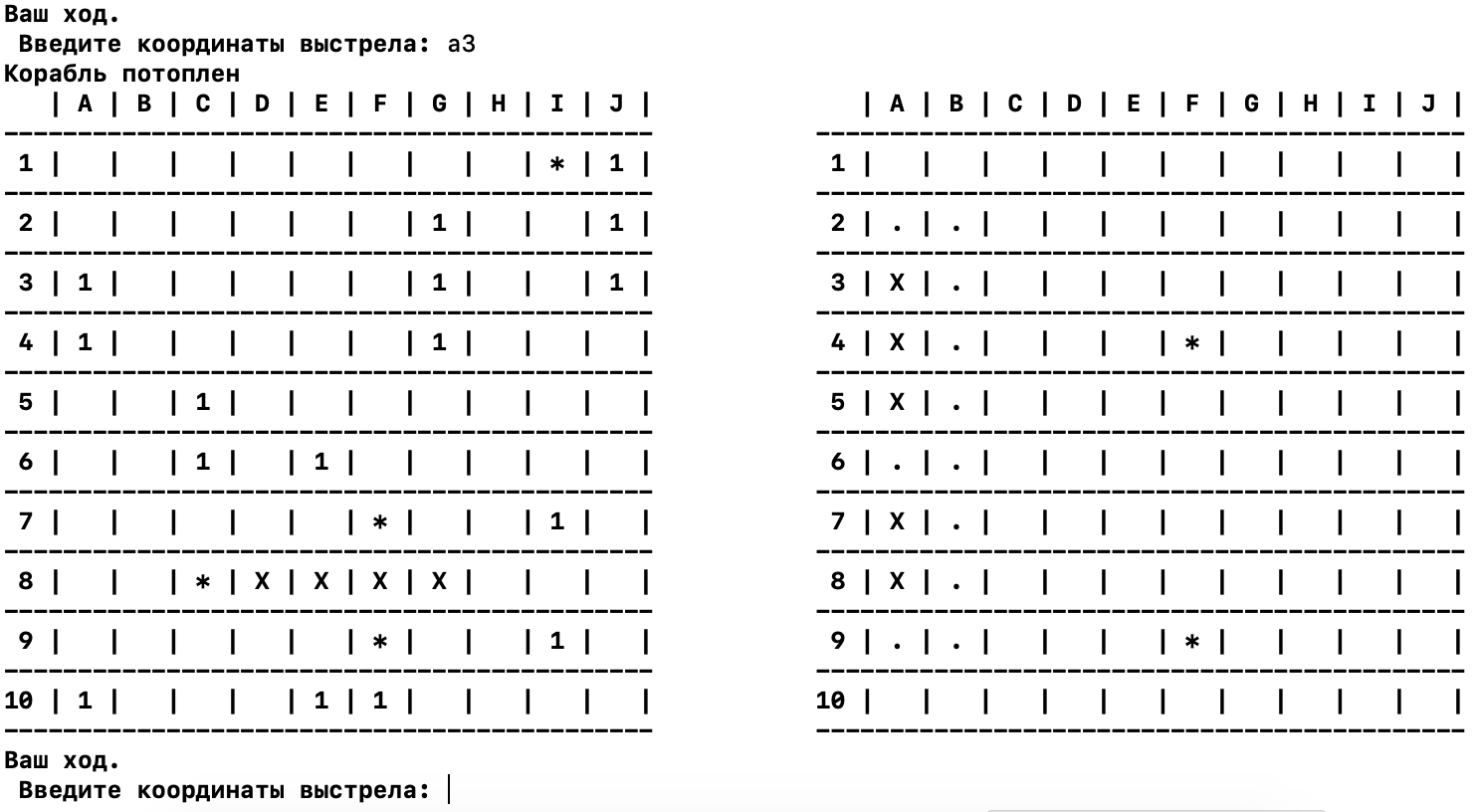


Рисунок 5

При ходе компьютера на экран выводятся координаты и результат выстрела, чтобы игроку было проще следить за ходом игры. После первого промаха компьютера ход переходит к игроку, при этом на экране появляется фраза ‘‘Ваш ход.’’(Рисунок 6).

В случае введения координат, по которым уже был произведен выстрел, на экране появляется сообщение ‘‘Вы уже стреляли по этим координатам.’’. В случае введения координат, которые находятся вне игрового поля, и случае, описанном ранее, выводится сообщение ‘‘Вы ввели неверные координаты.’’. (см. Рисунок 7)

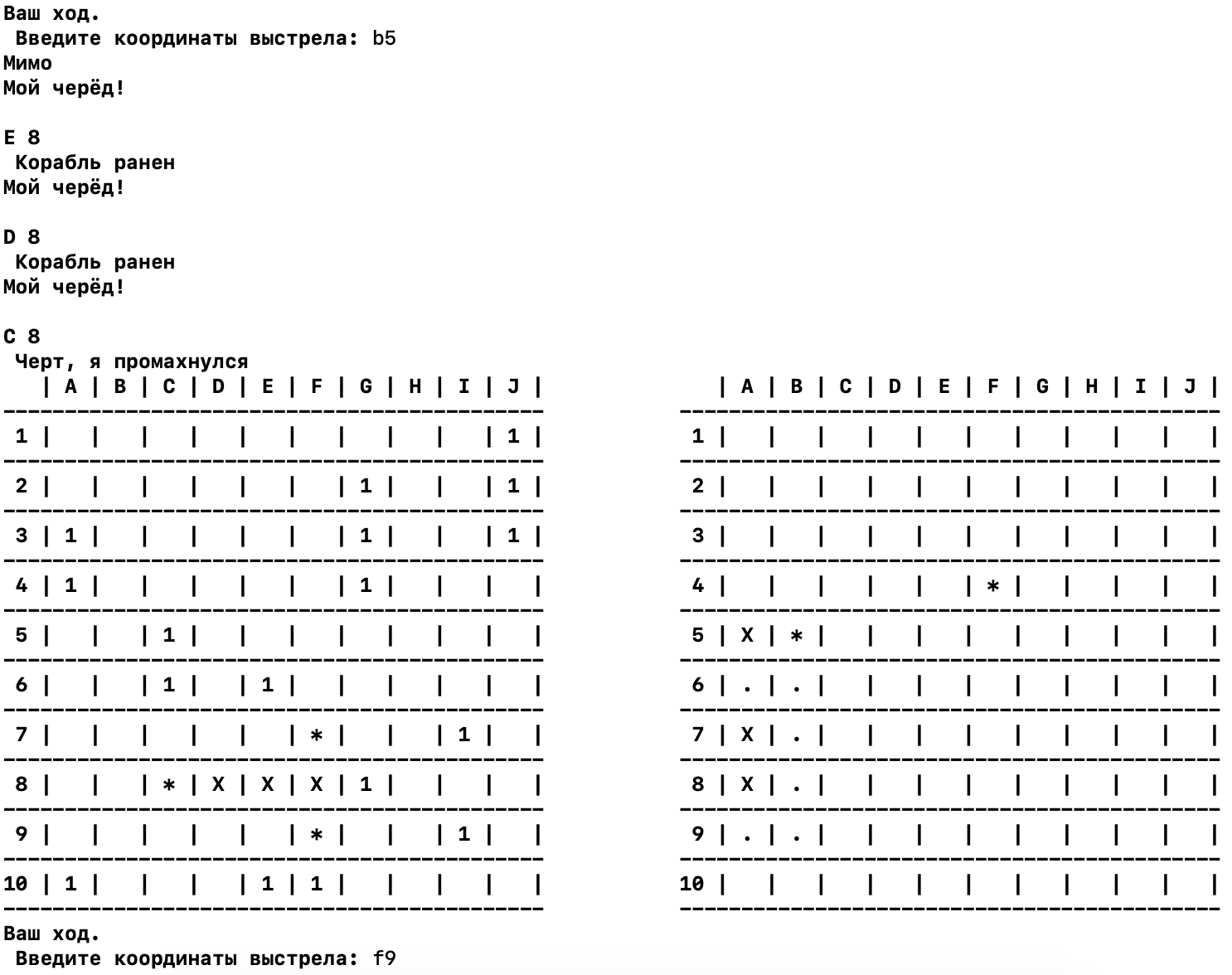


Рисунок 6



Рисунок 7

## Конец игры

При ликвидации всех кораблей компьютера, выводится сообщение ‘‘Вы выиграли! Поздравляю!’’(см. Рисунок 8). В случае, если победил компьютер, на экране появляется сообщение ‘‘К сожалению, выиграл компьютер. Не расстраивайте, вы можете взять реванш!’’.

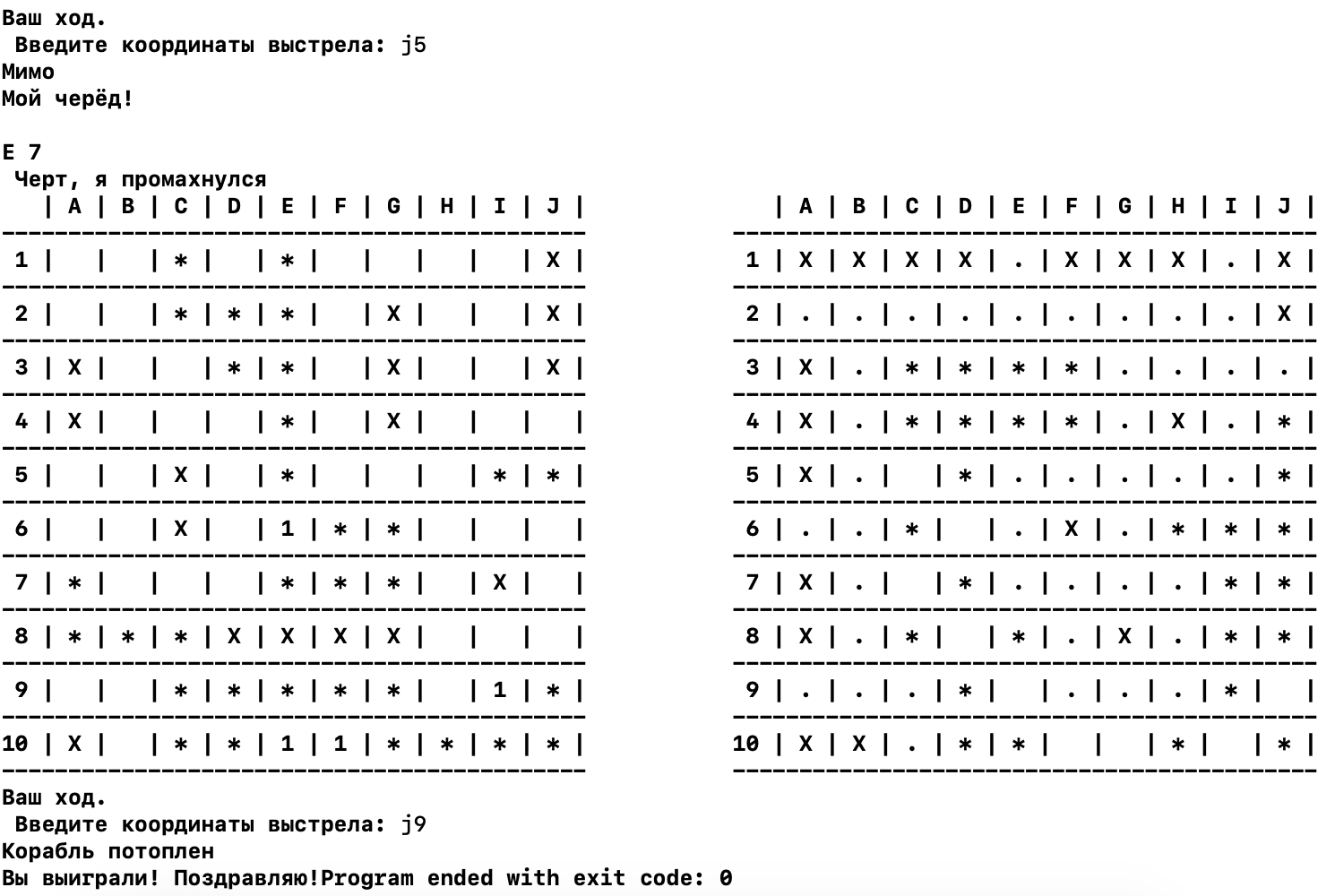


Рисунок 8

# ЗАКЛЮЧЕНИЕ

В процессе выполнения курсового проекта были получены навыки проектирования консольной игры и разработки искусственного интеллекта. Была реализована консольная игра “Морской бой”, удовлетворяющая всем правилам классической версии игры.