

«Морской бой»

Создание компьютерных игр во внеурочной
деятельности

Реферат выступления Фомина А. Т. учителя информатики гимназии №14
им. Ю. А. Гагарина г. Ейска МО Ейский район, на заседании РМО от 23.08.2018 г.
Опубликована сайте http://inf-w.ru/?page_id=7752
(Верстка брошюрой)

Последняя правка: 23.08.18

Пожелания и замечания направлять по адресу: sakhastro@mail.ru

Оглавление

| | |
|-----------------------------------------------|----|
| Идея..... | 3 |
| Введение..... | 3 |
| Выбор игры и стратегии..... | 6 |
| Этапы проектной работы..... | 8 |
| Реализация..... | 11 |
| Исходный код..... | 11 |
| Краткое описание функций и данных..... | 25 |
| Ввод/вывод программы в текстовой консоли..... | 28 |
| Заключение..... | 42 |
| Литература..... | 43 |

Идея

Программист не играет в игры, он — создает их.

Введение

Кто из нас не играл в компьютерные игры? Для кого-то компьютерная игра просто хобби, кто-то маниакально зависим от игр (и мы это не приветствуем), некоторые используют компьютерные игры для веселого время препровождения, чтобы отвлечься и расслабиться, а кто-то использует игры для обучения — и это мы, учителя!

Игровое обучение имеет глубокие исторические корни. Известно насколько игра многогранна. Она обучает, развивает, воспитывает, социализирует, развлекает и дает отдых. Но сегодня мы поговорим не об игровом методе обучения (хотя и он может применяться с пользой во время проведения внеурочной работы в компьютерном классе). Речь пойдет о создании компьютерной игры — как о большом проекте, который может быть реализован в кружковой работе, в виде электива или на факультативных занятиях.

Идеи для проектных работ можно почерпнуть в популярной компьютерной литературе для детей и начинающих разработчиков. Для чтения этих книг какой бы то ни было специальной подготовки не требуется. Заслуживают внимания работы следующих авторов: У. Сэнд, К. Сэнд Hello World! Занимательное программирование. — СПб.: Питер, 2016. (Серия «Вы и ваш ребенок»), Доусон М. Изучаем С++ через программирование игр. - СПб.: Питер, 2016 и Васильев А. Н. Самоучитель С++ с примерами и задачами. — СПб.: Наука и Техника, 2016. Но в познавательной сфере обучающихся должна обязательно находиться и литература развивающая умения составления сложных алгоритмов. Среди источников заслуживающих доверия наибольшей популярностью пользуются работы Скиена С. Алгоритмы. Руководство по разработке. СПб.: БХВ-Петербург, 2011, Дасгупта С. и др. Алгоритмы / С. Дасгупта, Х. Пападимитриу, У. Вазирани. М.: МЦНМО, 2014 и Хайнеман Д., Поллис Г., Селков С. Алгоритмы. Справочник с примерами на С, С++, Java и Python. СПб.: ООО «Альфа-книга», 2017. Знание классических алгоритмов совершенно необходимо для составления

эффективных программ, в том числе игровых.

Когда появились первые компьютерные игры? Первые игровые программы были созданы для первых компьютеров, которые, конечно, предназначались для совершенно иных целей. Как полагают, исторически первыми компьютерными игровыми программами были «OXO» (имитация игры «крестики-нолики»), составленная для компьютера EDSAC британским ученым А. Дугласом (1952) и игровая программа «Spacewar!» написанная С. Расселом и др. студентами Массачусетского технологического института для мини-компьютера PDP-1 (Programmed Data Processor-1, 1962).

Известен интересный факт из истории компьютерных игр, который послужил толчком к развитию системного программного обеспечения. В 1969 г. Кен Томпсон, известный пионер информационных технологий (сотрудник Bell Labs), написал игру Space Travel. Томпсон и Деннисом Ритчи портировали эту игру на PDP-7, который, в то время, обладал хорошим графическим терминалом. Процесс портирования осложнялся тем, что у программистов не было никаких утилит для составления программ.

Специально для этой игры, им пришлось написать на ассемблере библиотеку, графическую подсистему и отладчик. Об этом интересном факте умалчивают многие учебники информатики. А между тем, составленный ими исходный код, в конечном итоге, превратился в операционную систему Unix. [16]

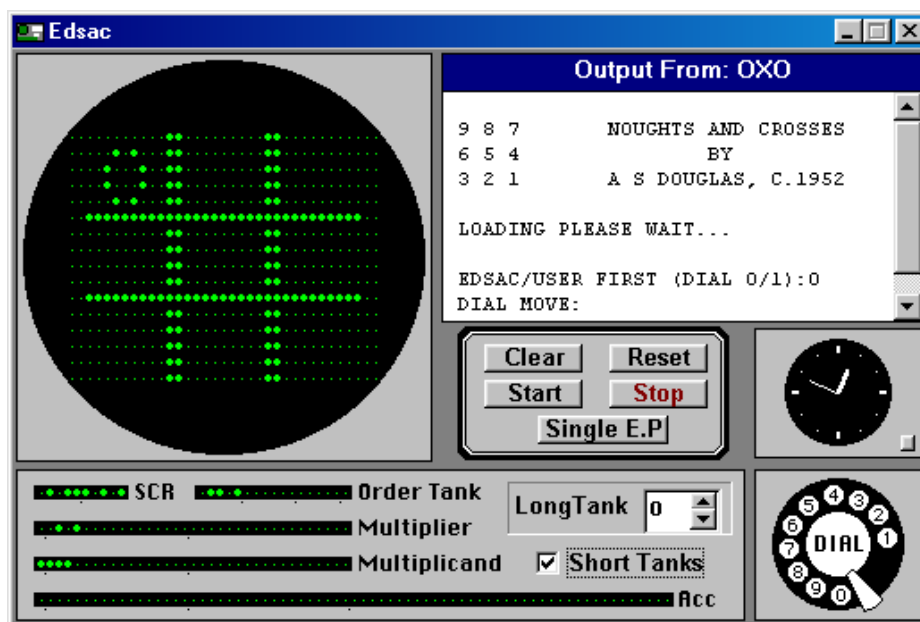


Рис. 1. «OXO» запущенная в эмуляторе



Рис. 2. Пионеры IT Кен Томпсон (слева) и Денис Ритчи



Рис. 3. Компьютер PDP-7

Выбор игры и стратегии

Итак, разработка компьютерных игр стимулировала развитие важной отрасли информационных технологий — системное программирование. Но сможем ли мы достичь ощутимых педагогических результатов во внеурочной деятельности разрабатывая компьютерные игры? Положительный ответ на этот вопрос будет складываться из нескольких факторов. Самый главный из них — уровень развития обучающихся. Разработка алгоритма потребует хороших знаний программирования (местами выходящего за рамки школьной программы) и математических основ информатики. Второй фактор — разумные временные рамки проекта. Поддержка мотивации на должном уровне требует осязаемости текущих и перспективных результатов работы. Третий фактор — умение и желание работать в команде. Это, пожалуй, самое «слабое звено». Создать дружественную атмосферу сотрудничества — непростая задача для педагога.

Поисковая деятельность и применение всех, доступных обучающимся, инструментов языка программирования, работа с источниками и привлечение знаний из других областей науки и техники позволит решить главную задачу внеурочной деятельности — установить прочную связь теории и практики, связь, которую трудно удерживать в течение длительного и сложного курса алгоритмизации и программирования, но сравнительно легко при работе над большим проектом.

Началом работы над проектом является целеполагание — выбор игры и стратегии реализации. Здесь мы делаем ставку на консольные игры (не путать с игровыми консолями), т. е. игры, в которых результаты выводятся в текстовой консоли на экране монитора. Для таких игр не нужна сложная реализация графических интерфейсов, а выбор игры будет зависеть только от уровня развития обучающихся. Младшие школьники могут создавать очень простые игры, без вспомогательных алгоритмов (в пределах главной функции программы). Программы этих игр не обязательно разбивать на подзадачи. Примерами таких игр являются программы «Тест по таблице умножения» (и прочие тесты), «Угадай число», игра в кости и даже примитивная реализация «Крестики-нолики».

При решении более сложных задач рамки одной функции становятся тесными. Появляется необходимость разбиения решения задачи на отдельные подзадачи. Учащиеся переходят в своем развитии «на уровень вверх» и начинают знакомиться с парадигмами программирования. Первая парадигма (или стиль) написания программ — это процедурное программирование. Рассматриваемая в настоящей работе программа «Морской бой» принадлежит именно к этой категории разработки.

Использование в программе функций позволяет преподавателю организовать коллективную разработку алгоритма. Сначала прорабатываются теоретические основы игровой стратегии. Строится упрощенная блок-схема будущей игры.

Определяются глобальные объекты (переменные и структуры данных), позволяющие увязать все функции в единой программе. Затем выявляются процедуры, которые должны разрабатываться в виде отдельных алгоритмов (подзадач). Поскольку подзадачи могут иметь различную сложность они могут решаться как индивидуально, так и в группах. На заключительном этапе разрабатывается главная функция программы, которую тоже можно составить коллегиально. Ярким примером игр, которые создаются с помощью процедурного программирования являются карточные игры¹. [1,358]

Следующий уровень в сложности разработки компьютерных игр — это использование Объектно-ориентированного подхода (ООП). Почему нужно становиться объектно-ориентированным? Дело в том, что объекты позволяют собрать во едино и функции, и данные. Поэтому, в отличие от функций, объект может хранить состояние, а это очень важно при разработках сложных игровых программ, поскольку позволяет абстрактно управлять объектами (изменяя их свойства). [4,154]

Но переход на этот уровень сопряжен с проблемой. Она заключается в том, что программирование в стиле ООП на базовом уровне не изучается вовсе, а на профильном уровне не изучается полноценно и всесторонне в рамках изучения конкретного языка программирования. Но так ли уж нужно изучать ООП в школе? На базовом уровне — никакой необходимости, конечно, нет. Поскольку это всего-лишь стиль. Даже на олимпиадах вполне можно обойтись процедурным стилем. Но объектно-ориентированное программирование позволяет решать задачи более эффективно, а сопровождать и документировать программы гораздо проще.

Выбор такого подхода могут сделать высокомотивированные обучающиеся, которые увлекаются программированием более серьезно и со значительным опережением программы. Замес из разных стилей не мешает коллективному творчеству в составлении программы в которой определенные её части могут быть реализованы с применением различных подходов. Благо, используемый нами язык программирования C++ позволяет это сделать с максимальным комфортом.

1 При составлении программ «карточных игр» с равным успехом можно использовать и ООП-стиль.

Этапы проектной работы

Для описания примера проектной работы по созданию компьютерных игр я выбрал игру «Морской бой». Эта 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. [7]

В нашей версии противником будет являться — компьютер, который должен самостоятельно расставить корабли на своей виртуальной карте (скрытой от игрока), перехватывать инициативу и осуществлять очередной ход. При этом игрок вручную вводит координаты кораблей, расставляя их на своей карте согласно правил, и координаты очередного хода (или выстрела).

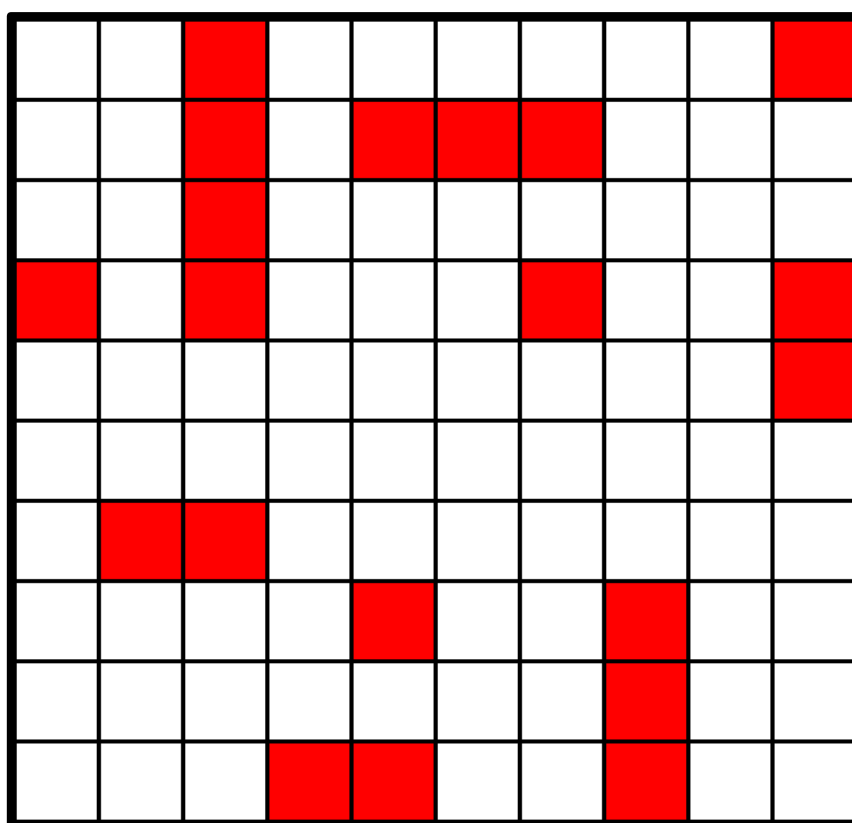


Рис. 4. Возможный вид карты игрока

Игровое поле (Рис. 1) представляет собой квадрат размером 10x10. На этом поле размещаются корабли:

- 1 корабль — ряд из 4 клеток («четырёхпалубный»; линкор)
- 2 корабля — ряд из 3 клеток («трёхпалубные»; крейсера)
- 3 корабля — ряд из 2 клеток («двухпалубные»; эсминцы)
- 4 корабля — 1 клетка («однопалубные»; торпедные катера)

Корабли не должны касаться друг друга сторонами и углами². Такую версию игры мы назовем - «классическая». Обращаем ваше внимание на важный факт (используемый в предложенной программе), что всего на поле будет занято ровно 20 клеток.

Система координат применяемая при разработке игры: центр координат левый верхний угол, ось X — по горизонтали, ось Y — по вертикали (положительный отсчет «вниз») (Рис. 2). Стоит обратить внимание обучающихся при моделировании полей, что в матрице ось X соответствует индексу «j», а Y — индексу «i», тем самым, предупредив возможную путаницу. Как это ни странно работа с координатами наиболее трудоемкая часть программы.

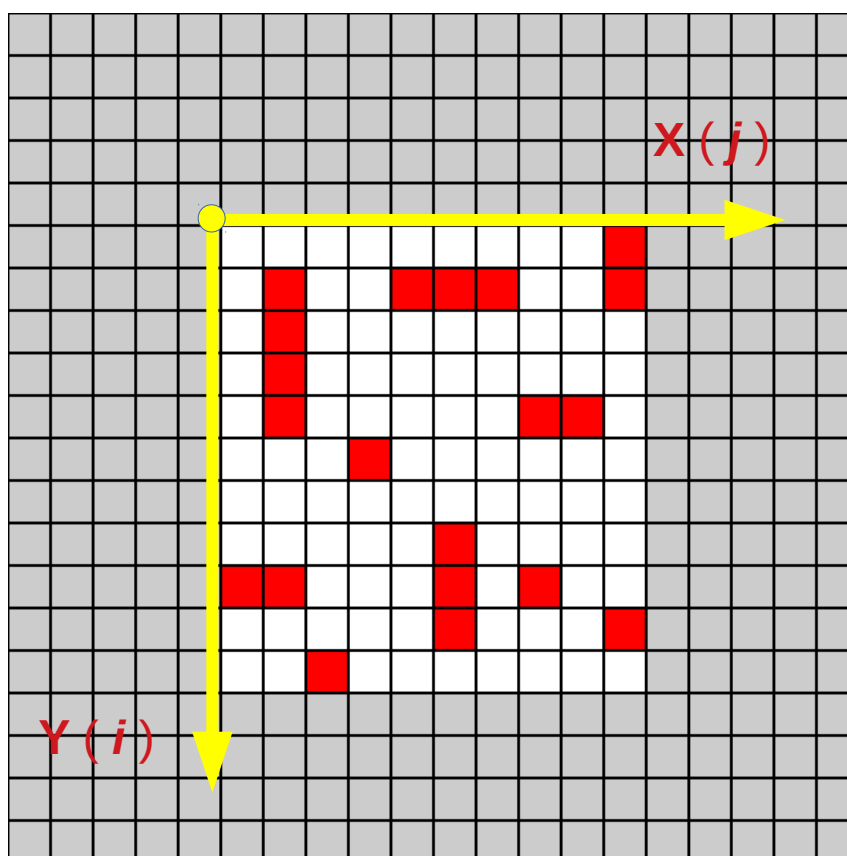


Рис. 5. Система координат

Начало проектной работы, как мы уже сказали, — это построение блок-схемы будущей игры и выделение в алгоритме отдельных программных блоков, которые будут разрабатывать обучающиеся самостоятельно. В предложенной программе насчитывается 12 функций и процедур:

1. Определение доступности места размещения;
2. Получения координат;
3. Создание карты компьютера;

² На самом деле, существуют, помимо классической, и другие версии игры в которых данные правила нарушаются.

4. Создание карты игрока;
5. Рисование корабля;
6. Инициализация карт;
7. Вывод карты игрока;
8. Вывод карты компьютера;
9. Морская баталия;
10. Выстрел игрока;
11. Выстрел компьютера;
12. Получение координат для компьютера.

Ниже мы дадим отдельную характеристику наиболее важным и сложным функциям из этого перечня (в языке С++ и процедуры и функции называются одинаково — функции).

При проектировании компьютерной игры можно подойти к организации работы иначе: не поручать разработку отдельных блоков разным учащимся, а позволить каждому юному разработчику самому (или с помощью товарищей) подойти к реализации одной и той же общей задачи. И поэтапно разрабатывать каждый программный блок коллективно или индивидуально, но в сотрудничестве и под руководством учителя.

После того, как программа будет разработана и протестирована (а это является немаловажной частью проектной работы) можно провести обсуждение «узких» мест готового продукта на предмет их совершенствования. Пусть ребята предложат свои решения или идеи по изменению программного кода в лучшую сторону.

Можно, наконец, предложить свою версию игры «Морской бой». Например, на карту можно добавить мину или несколько мин, определив соответствующие правила при попадании на мину. Или увеличить размер поля, скажем, до 15 или 20 «клеток», одновременно увеличив размер и число кораблей. Также можно добавить диагональную ориентацию кораблей и ввести еще одного участника — подводную лодку.

Несложно заметить, что в предложенной программе победителем гарантированно будет человек. Дело в том, что компьютеру не хватает «интеллекта», чтобы сражаться с игроком. Выстрел, который делает компьютер — это случайный набор координат. Следовательно, не выделяется «ореол» вокруг пораженных клеток, не осуществляется возможный «прострел» на выявление многопалубного корабля. Наконец, не учитывается существующая выигрышная стратегия в классическом «Морском бое». Решение этой интересной задачи можно предложить ученикам на следующем этапе разработки ПО.

Заключительным этапом работы может стать общешкольный турнир, который может быть приурочен к неделе информатики, с награждением как победителей, так и разработчиков этой компьютерной игры.

Реализация

Исходный код

```
1. //=====
2. // Name      : morskoy_boy.cpp
3. // Author    : Fomin Andrew T.
4. // Version   : 0.1
5. // Copyright : GPL v.3
6. // Description : Консольная игра "Морской бой"
7. //=====
8. #include <iostream>
9. #include <iomanip>
10. #include <vector>
11. #include <random>
12. #include <ctime>
13. using namespace std;
14. // Прототипы для создания карты
15. // Определение доступности места размещения
16. bool test(vector<vector<bool>>&,
17.           pair<int, int>&,
18.           int, bool&);
19. // Получение координат
20. pair<int, int> coord(int&, bool&);
21. // Создание карты
22. void mapSea(vector<vector<bool>>&, bool&);
23. // Рисование корабля
24. void ship(vector<vector<bool>>&,
25.           int&, bool&,
26.           pair<int, int>&);
27.
28. // Инициализация карт
29. void mapsIni(vector<vector<bool>>&,
30.             vector<vector<char>>&,
31.             vector<vector<bool>>&,
32.             vector<vector<char>>&);
33. // Создание карты игрока
```

```

34. void mapSeaUserCreat(vector<vector<bool>>&,
35.                      vector<vector<char>>&,
36.                      vector<vector<bool>>&,
37.                      vector<vector<char>>&,
38.                      bool&);
39. // Вывод карты игрока
40. void printMapUser(vector<vector<char>>&,
41.                  vector<vector<bool>>&,
42.                  bool&);
43. // Вывод карты компьютера
44. void printMapComp(vector<vector<char>>&);
45.
46. // battle
47. void battle(vector<vector<bool>>&,
48.            vector<vector<char>>&,
49.            vector<vector<bool>>&,
50.            vector<vector<char>>&);
51. // выстрел игрока
52. void shot_user(vector<vector<bool>>&,
53.               vector<vector<char>>&,
54.               int&, bool&, pair<int, int>&);
55. // выстрел компьютера
56. void shot_comp(vector<vector<bool>>&,
57.               vector<vector<char>>&,
58.               int&,
59.               bool&);
60. // Определение координат для компьютера
61. pair<int, int> crdcomp();
62.
63. int main() {
64.     vector<vector<bool>> SEA(10);
65.     vector<vector<bool>> USER(10);
66.     vector<vector<char>> SeaVision(10);
67.     vector<vector<char>> UserVision(10);
68.     bool flag = false; // Подготовка
69.     // Вызываем главную функцию
70.     mapSeaUserCreat(USER,
71.                    UserVision,
72.                    SEA,
73.                    SeaVision,
74.                    flag);

```

```

75.     // Началась игра
76.     return 0;
77. }
78.
79. pair<int, int> coord(int &size, bool &f) {
80.     // Сначала определяем ориентацию
81.     // 0 - горизонт, 1 - вертикаль
82.     static default_random_engine rnd(time(0));
83.     static uniform_int_distribution<int> t(0, 1);
84.     f = bool(t(rnd));
85.     f = bool(t(rnd));
86.     static uniform_int_distribution<int> d(0, 9);
87.     int x = d(rnd);
88.     x = d(rnd);
89.     int y = d(rnd);
90.     // Если вышли за пределы карты разворачиваем
91.     // по горизонтали двигаем влево, так чтобы
92.     // правым бортом оперся о край карты (нос слева)
93.     if (!f && x + size > 10) x = 10 - size;
94.     // по вертикали тоже, но опираем задней частью
95.     // нос вверху
96.     if (f && y + size > 10) y = 10 - size;
97.     pair<int, int> xy = make_pair(x, y);
98.     return xy;
99. }
100. bool test(vector<vector<bool>> &sea,
101.           pair<int, int> &crd,
102.           int size,
103.           bool &f) {
104.     int x = crd.first, y = crd.second;
105.     bool b = true;
106.     // Обратить внимание на то, что i -> y, а j -> x
107.     // Горизонтально
108.     if (!f) {
109.         // не боковые и не угловые
110.         if (x > 0 && y > 0 &&
111.             x < 10 - size &&
112.             y < 9) {
113.             for (int i = y - 1; i <= y + 1; i++)
114.                 for (int j = x - 1; j <= x + size; j++)
115.                     if (sea[i][j] == 1)

```

```

116.             b = false;
117.         // угловые
118.     } else
119.         if (x == 0 && y == 0) {
120.             for (int i = y; i <= y + 1; i++)
121.                 for (int j = x; j <= x + size; j++)
122.                     if (sea[i][j] == 1)
123.                         b = false;
124.     } else
125.         if (x == 10 - size && y == 0) {
126.             for (int i = y; i <= y + 1; i++)
127.                 for (int j = x; j <= x + size - 1;
128.                     j++)
129.                     if (sea[i][j] == 1)
130.                         b = false;
131.     } else
132.         if (x == 10 - size && y == 9) {
133.             for (int i = y - 1; i <= y; i++)
134.                 for (int j = x - 1;
135.                     j <= x + size - 1; j++)
136.                     if (sea[i][j] == 1)
137.                         b = false;
138.     } else
139.         if (x == 0 && y == 9) {
140.             for (int i = y - 1; i <= y; i++)
141.                 for (int j = x; j <= x + size; j++)
142.                     if (sea[i][j] == 1)
143.                         b = false;
144.     } else
145.     // Не угловые по бокам
146.         if (x > 0 && x < 10 - size && y == 0) {
147.             for (int i = y; i <= y + 1; i++)
148.                 for (int j = x - 1; j <= x + size;
149.                     j++)
150.                     if (sea[i][j] == 1)
151.                         b = false;
152.     } else
153.         if (y > 0 && y < 9 && x == 10 - size) {
154.             for (int i = y - 1; i <= y + 1; i++)
155.                 for (int j = x - 1;
156.                     j <= x + size - 1; j++)

```

```

157.             if (sea[i][j] == 1)
158.                 b = false;
159.         } else
160.             if (x > 0 && x < 10 - size && y == 9) {
161.                 for (int i = y - 1; i <= y; i++)
162.                     for (int j = x - 1; j <= x + size;
163.                         j++)
164.                         if (sea[i][j] == 1)
165.                             b = false;
166.             } else
167.                 if (y > 0 && y < 9 && x == 0) {
168.                     for (int i = y - 1; i <= y + 1; i++)
169.                         for (int j = x; j <= x + size; j++)
170.                             if (sea[i][j] == 1)
171.                                 b = false;
172.                 }
173.         } else {
174.             // Вертикально
175.             // не боковые и не угловые
176.             if (x > 0 && y > 0 && x < 9 && y < 10 - size) {
177.                 for (int i = y - 1; i <= y + size; i++)
178.                     for (int j = x - 1; j <= x + 1; j++)
179.                         if (sea[i][j] == 1)
180.                             b = false;
181.             // угловые
182.         } else
183.             if (x == 0 && y == 0) {
184.                 for (int i = y; i <= y + 1; i++)
185.                     for (int j = x; j <= x + size; j++)
186.                         if (sea[i][j] == 1)
187.                             b = false;
188.         } else
189.             if (x == 9 && y == 0) {
190.                 for (int i = y; i <= y + 1; i++)
191.                     for (int j = x - 1; j <= x; j++)
192.                         if (sea[i][j] == 1)
193.                             b = false;
194.         } else
195.             if (x == 9 && y == 10 - size) {
196.                 for (int i = y - 1; i <= y + size - 1; i++)
197.                     for (int j = x - 1; j <= x; j++)

```

```

198.             if (sea[i][j] == 1)
199.                 b = false;
200.         } else
201.             if (x == 0 && y == 10 - size) {
202.                 for (int i = y - 1; i <= y + size - 1; i++)
203.                     for (int j = x; j <= x + 1; j++)
204.                         if (sea[i][j] == 1)
205.                             b = false;
206.             } else
207.             // не угловые по бокам
208.             if (x > 0 && x < 9 && y == 0) {
209.                 for (int i = y; i <= y + size; i++)
210.                     for (int j = x - 1; j <= x + 1; j++)
211.                         if (sea[i][j] == 1)
212.                             b = false;
213.             } else
214.             if (y > 0 && y < 10 - size && x == 9) {
215.                 for (int i = y - 1; i <= y + size; i++)
216.                     for (int j = x - 1; j <= x; j++)
217.                         if (sea[i][j] == 1)
218.                             b = false;
219.             } else
220.             if (x > 0 && x < 9 && y == 10 - size) {
221.                 for (int i = y - 1; i <= y + size - 1; i++)
222.                     for (int j = x - 1; j <= x + 1; j++)
223.                         if (sea[i][j] == 1)
224.                             b = false;
225.             } else
226.             if (y > 0 && y < 10 - size && x == 0) {
227.                 for (int i = y - 1; i <= y + size; i++)
228.                     for (int j = x; j <= x + 1; j++)
229.                         if (sea[i][j] == 1)
230.                             b = false;
231.             }
232.         }
233.     return b;
234. }
235. void ship(vector<vector<bool>> &sea,
236.           int &size,
237.           bool &f,
238.           pair<int, int> &crd) {

```



```

239.     int x = crd.first;
240.     int y = crd.second;
241.     if (!f)
242.         for (int i = x; i < x + size; i++)
243.             sea[y][i] = 1;
244.     else
245.         for (int i = y; i < y + size; i++)
246.             sea[i][x] = 1;
247. }
248. void mapSea(vector<vector<bool>> &sea, bool &f) {
249.     // Заполняем карту компьютера
250.     // Координаты и ориентация (передается по ссылке bool f)
251.     pair<int, int> xy;
252.     // Начинаем с самого большого, потом по нисходящей
253.     // Размер
254.     //===== 4 =====
255.     int size = 4;
256.     xy = coord(size, f);
257.     ship(sea, size, f, xy);
258.
259.     //===== 3 =====
260.     size--;
261.     int j = 0; // Здесь уже счетчиком
262.     bool q; // Для лучшего восприятия кода
263.     do {
264.         xy = coord(size, f);
265.         q = test(sea, xy, size, f);
266.         if (q) {
267.             j++;
268.             ship(sea, size, f, xy);
269.         }
270.     } while (j != 2);
271.
272.     //===== 2 =====
273.     size--;
274.     j = 0;
275.     do {
276.         xy = coord(size, f);
277.         q = test(sea, xy, size, f);
278.         if (q) {
279.             j++;

```

```

280.             ship(sea, size, f, xy);
281.         }
282.     } while (j != 3);
283.
284.     //===== 1 =====
285.     size--;
286.     j = 0;
287.     do {
288.         xy = coord(size, f);
289.         q = test(sea, xy, size, f);
290.         if (q) {
291.             j++;
292.             ship(sea, size, f, xy);
293.         }
294.     } while (j != 4);
295. }
296. //=====INI=====
297. void mapsIni(vector<vector<bool>> &usermapbool,
298.             vector<vector<char>> &uservision,
299.             vector<vector<bool>> &compmapbool,
300.             vector<vector<char>> &compvvision) {
301.     for (size_t i = 0; i < 10; i++)
302.         for (size_t j = 0; j < 10; j++) {
303.             uservision[i].push_back('.');
304.             compvvision[i].push_back('.');
305.         }
306.     for (size_t i = 0; i < 10; i++)
307.         for (size_t j = 0; j < 10; j++) {
308.             usermapbool[i].push_back(0);
309.             compmapbool[i].push_back(0);
310.         }
311. }
312. void printMapUser(vector<vector<char>> &uservision,
313.                  vector<vector<bool>> &usermapbool,
314.                  bool &check) {
315.     if (check) {
316.         // Значки:
317.         // # - контур корабля
318.         // X - попадание
319.         // . - море
320.         // 0 - промах

```

```

321.         for (int i = 0; i < 10; i++)
322.             for (int j = 0; j < 10; j++)
323.                 if (usermapbool[i][j] == 1)
324.                     uservision[i][j] = '#';
325.     cout << " ";
326.     for (int i = 1; i < 11; i++) cout << setw(3) << i;
327.     cout << "\n";
328.     for (int i = 0; i < 9; i++) {
329.         cout << i + 1 << setw(4);
330.         for (int j = 0; j < 10; j++) {
331.             cout << uservision[i][j] << setw(3);
332.         }
333.         cout << "\n";
334.     }
335.     cout << "10 ";
336.     for (int j = 0; j < 10; j++)
337.         cout << uservision[9][j] << setw(3);
338.     cout << endl;
339. } else {
340.     cout << " ";
341.     for (int i = 1; i < 11; i++) cout << setw(3) << i;
342.     cout << "\n";
343.     for (int i = 0; i < 9; i++) {
344.         cout << i + 1 << setw(4);
345.         for (int j = 0; j < 10; j++) {
346.             cout << uservision[i][j] << setw(3);
347.         }
348.         cout << "\n";
349.     }
350.     cout << "10 ";
351.     for (int j = 0; j < 10; j++)
352.         cout << uservision[9][j] << setw(3);
353.     cout << endl;
354. }
355. }
356. void printMapComp(vector<vector<char>> &compvision) {
357.     cout << " ";
358.     for (int i = 1; i < 11; i++) cout << setw(3) << i;
359.     cout << "\n";
360.     for (int i = 0; i < 9; i++) {
361.         cout << i + 1 << setw(4);

```

```

362.         for (int j = 0; j < 10; j++) {
363.             cout << compvision[i][j] << setw(3);
364.         }
365.         cout << "\n";
366.     }
367.     cout << "10  ";
368.     for (int j = 0; j < 10; j++)
369.         cout << compvision[9][j] << setw(3);
370.     cout << endl;
371. }
372. //=====
373. void mapSeaUserCreat(vector<vector<bool>> &usermapbool,
374.                     vector<vector<char>> &uservision,
375.                     vector<vector<bool>> &compmapbool,
376.                     vector<vector<char>> &compvision,
377.                     bool &check) {
378.     cout << "Добро пожаловать в игру\n"
379.          <<  "\"Морской бой\!"
380.          <<  "\nВы должны расставить свои корабли."
381.          <<  "\nГотовы?\nДа - 1\nНет - 0\n => ";
382.     bool f;
383.     cin >> f;
384.     if (!f) {
385.         cout << "Вы вышли!" << endl;
386.         exit(0);
387.     } else {
388.         mapsIni(usermapbool,
389.                uservision,
390.                compmapbool,
391.                compvision);
392.     }
393.     cout << "\nВедите координаты носов кораблей:\n"
394.          <<  "горизонтальные - нос слева\n"
395.          <<  "вертикальные - нос сверху;\n"
396.          <<  "начало координат левый верхний угол."
397.          <<  "\nКарта-подсказка:\n" << endl;
398.     printMapUser(uservision, usermapbool, check);
399.     cout << "\nВведите координаты (x, y) для 10 кораблей:\n";
400.     cout << " X Y" << endl;
401.     int x, y;
402.     int k = 0;

```

```

403.     int size;
404.     for (int i = 1; i < 5; i++) {
405.         size = 5 - i;
406.         for (int j = 1; j <= i; j++) {
407.             cout << ++k
408.                 << " => "
409.                 << size
410.                 << "-палубный\nx = ";
411.             cin >> x;
412.             cout << "y = ";
413.             cin >> y;
414.             --x; --y;
415.             pair<int, int> xy = make_pair(x, y);
416.             cout << "Вертикаль (1) или горизонт (0): ";
417.             cin >> f;
418.             ship(usermapbool, size, f, xy);
419.         }
420.     }
421.     check = true;
422.     cout << "Ваша карта:" << endl;
423.     printMapUser(uservision, usermapbool, check);
424.     // В этом месте можно использовать еще одну функцию
425.     // повторного ввода карты (если что-то не понравилось)
426.     // И создаем новую карту компьютера
427.     static default_random_engine rnd(time(0));
428.     static uniform_int_distribution<int> t(0, 1);
429.     check = bool(t(rnd));
430.     check = bool(t(rnd));
431.     mapSea(compmapbool, check);
432.     battle(usermapbool, uservision, compmapbool, compvision);
433. }
434. void battle(vector<vector<bool>> &usermapbool,
435.             vector<vector<char>> &uservision,
436.             vector<vector<bool>> &compmapbool,
437.             vector<vector<char>> &compvision) {
438.     int x, y;
439.     pair<int, int> xy;
440.     bool step;
441.     int usr_count = 20;
442.     int cmp_count = 20;
443.     do {

```

```

444.         step = true;
445.         while (step && usr_count) {
446.             cout << "Ваш ход!\n"
447.                 << "Определите цель:"
448.                 << "\nX = ";      cin >> x;
449.             cout << "Y = "; cin >> y;
450.             --x; --y;
451.             xy = make_pair(x,y);
452.             shot_user(compmapbool,
453.                     compvision,
454.                     usr_count,
455.                     step,
456.                     xy);
457.             if (step) {
458.                 cout << "Точное попадание!" << endl;
459.                 printMapComp(compvision);
460.             } else {
461.                 cout << "Вы не попали. Играет компьютер"
462.                     << endl;
463.             }
464.         }
465.         if (!usr_count) {
466.             cout << "Поздравляем, Вы победили!" << endl;
467.             break;
468.         }
469.         step = true;
470.         while (step) {
471.             shot_comp(usermapbool,
472.                     uservision,
473.                     cmp_count,
474.                     step);
475.             if (step) {
476.                 cout << "Компьютер попал!" << endl;
477.                 bool f = !step;
478.                 printMapUser(uservision, usermapbool, f);
479.             } else {
480.                 cout << "Компьютер не попал!" << endl;
481.             }
482.         }
483.     } while (usr_count && cmp_count);
484.     cout << "Хотите повторить?\n"

```

```

485.         << "Да - 1\nНет - 0\n => ";
486.     bool f;
487.     cin >> f;
488.     if (!f) {
489.         cout << "Вы вышли!" << endl;
490.         exit(0);
491.     } else {
492.         f = !f;
493.         mapsIni(usermapbool,
494.                 uservision,
495.                 compmapbool,
496.                 compvision);
497.         mapSeaUserCreat(usermapbool,
498.                          uservision,
499.                          compmapbool,
500.                          compvision,
501.                          f);
502.     }
503. }
504. pair<int, int> crdcomp() {
505.     static default_random_engine rnd(time(0));
506.     static uniform_int_distribution<int> d(0, 9);
507.     int x = d(rnd);
508.     x = d(rnd);
509.     int y = d(rnd);
510.     pair<int, int> xy = make_pair(x, y);
511.     return xy;
512. }
513. void shot_user(vector<vector<bool>> &compmapbool,
514.               vector<vector<char>> &compvision,
515.               int &usr_count, bool &usr_step,
516.               pair<int, int> &xy) {
517.     int x = xy.first;
518.     int y = xy.second;
519.     if (compmapbool[y][x] == 1) {
520.         usr_count--;
521.         compvision[y][x] = 'X';
522.     } else {
523.         compvision[y][x] = '0';
524.         usr_step = false;
525.     }

```

```
526. }
527. void shot_comp(vector<vector<bool>> &usermapbool,
528.               vector<vector<char>> &uservision,
529.               int &cmp_count,
530.               bool &usr_step) {
531.     pair<int, int> xy = crdcomp();
532.     int x = xy.first;
533.     int y = xy.second;
534.     if (usermapbool[y][x] == 1) {
535.         cmp_count--;
536.         uservision[y][x] = 'X';
537.     } else {
538.         uservision[y][x] = '0';
539.         usr_step = false;
540.     }
541. }
```


Краткое описание функций и данных

1. В главной функции `main()` определяются четыре двумерных массива: два булевских для фиксации расположения кораблей противников и два символьных для вывода карты промежуточных и итоговых результатов:

```
vector<vector<bool>> SEA(10);  
vector<vector<bool>> USER(10);  
vector<vector<char>> SeaVision(10);  
vector<vector<char>> UserVision(10);
```

А также булевский флаг для определения какой тип карты нужно выводить (образец или баталии)

```
bool flag = false;
```

Вызов функции `mapSeaUserCreat` включает последовательность вызовов всех остальных функций программы

```
mapSeaUserCreat(USER, UserVision, SEA, SeaVision, flag);
```

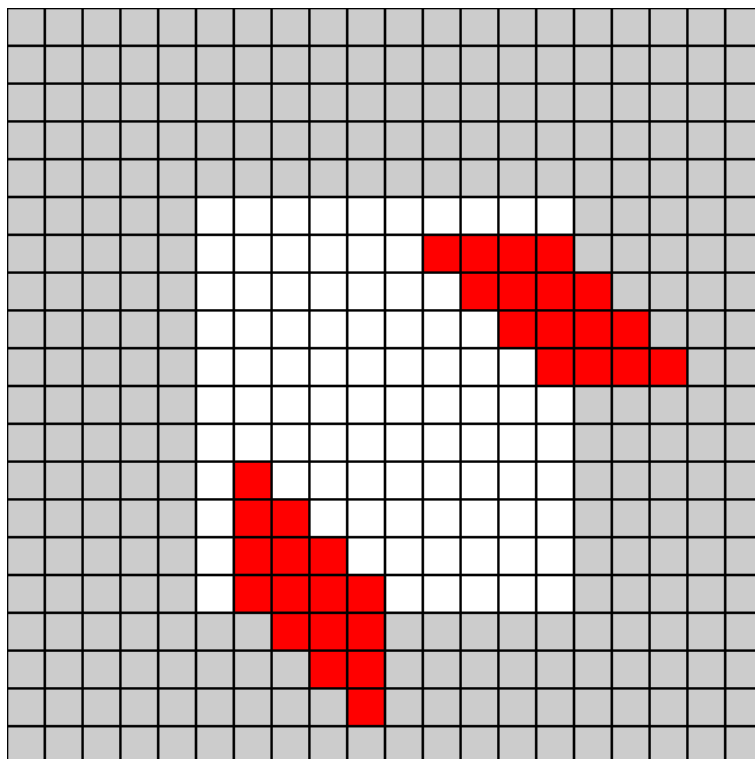


Рис. 6. Возможный выход кораблей за пределы карты

2. Назначение функции `mapSeaUserCreat` — это диалог с игроком, в результате которого инициализируются карты, вводятся координаты расположения кораблей игрока, выводятся карты подсказки и итоговой карты игрока (во время игры эти данные условно скрыты от компьютера, как

игрока), вызываются функции создания карт компьютера и следующая важная функция самой баталии. Эта функция не проверяет правильность расстановки кораблей игрока.

3. Функция `pair<int, int> coord(int &size, bool &f)` имеет два параметра `int size` — размер корабля и `bool f` — его ориентацию (горизонт или вертикаль), а возвращает пару координат, причем исправляется возможный выход корабля за границы карты (корабль будет сдвинут до границ влево или вверх) (Рис. 6).

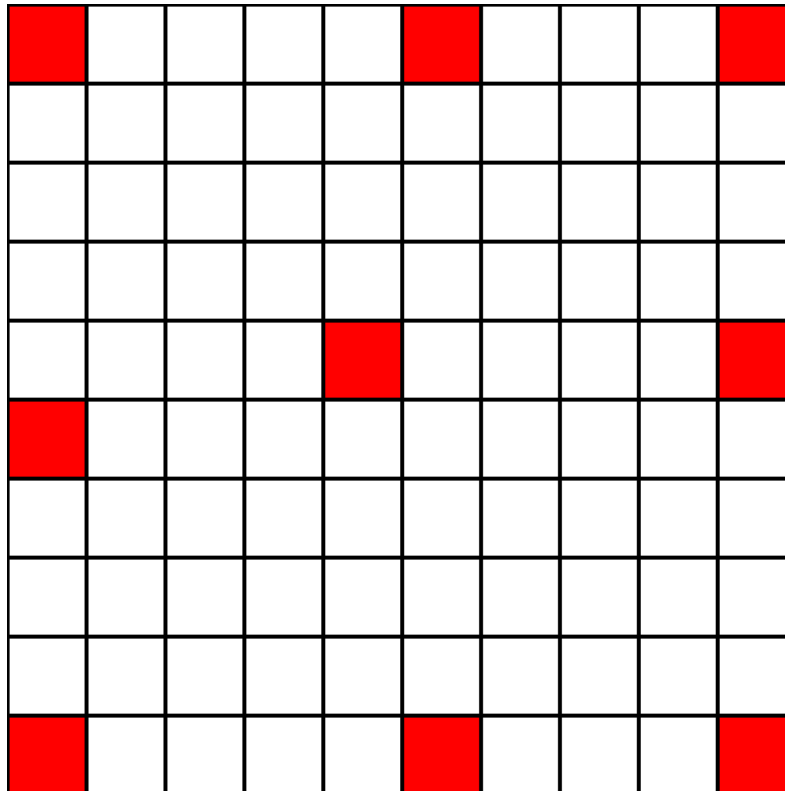


Рис. 7. Возможное положение судна на карте

4. Логическая функция

```
bool test(vector<vector<bool>> &sea, pair<int, int> &crd, int size, bool &f)
```

определяет возможность размещения корабля, согласно полученным координатам `pair<int, int> &crd`, длине `int size` и ориентации `bool &f`, учитывая возможные варианты расположения судна (Рис 7).

5. Функция

```
void ship(vector<vector<bool>> &sea, int &size, bool &f, pair<int, int> &crd)
```

строит корабль.

6. Функция

```
void mapSea(vector<vector<bool>> &sea, bool &f)
```

строит карту компьютера в такой последовательности: сначала размещается четырехпалубный в произвольном месте, все остальные, по нисходящей, с проверкой возможности такого размещения.

7. Последняя функция заслуживающая пояснения — это функция

```
void battle(vector<vector<bool>> &usermapbool,  
            vector<vector<char>> &uservision,  
            vector<vector<bool>> &compmapbool,  
            vector<vector<char>> &compvision)
```

Эта функция руководит самим сражением: производит выстрелы игрока и компьютера, запрашивает повторения боя или выход из программы, распечатывает карты игрока или компьютера в случае удачного попадания.

Ввод/вывод программы в текстовой консоли

Возможный вывод программы в текстовой консоли:

```
Добро пожаловать в игру
"Морской бой"!
Вы должны расставить свои корабли.
Готовы?
Да - 1
Нет - 0
=> 1

Ведите координаты носов кораблей:
горизонтальные - нос слева
вертикальные - нос сверху;
начало координат левый верхний угол.
Карта-подсказка:

    1  2  3  4  5  6  7  8  9 10
1   .  .  .  .  .  .  .  .  .  .
2   .  .  .  .  .  .  .  .  .  .
3   .  .  .  .  .  .  .  .  .  .
4   .  .  .  .  .  .  .  .  .  .
5   .  .  .  .  .  .  .  .  .  .
6   .  .  .  .  .  .  .  .  .  .
7   .  .  .  .  .  .  .  .  .  .
8   .  .  .  .  .  .  .  .  .  .
9   .  .  .  .  .  .  .  .  .  .
10  .  .  .  .  .  .  .  .  .  .

Введите координаты (x, y) для 10 кораблей:
X Y
1 => 4-палубный
x = 7
y = 2
Вертикаль (1) или горизонт (0): 1
2 => 3-палубный
x = 3
y = 9
Вертикаль (1) или горизонт (0): 0
3 => 3-палубный
x = 6
y = 10
Вертикаль (1) или горизонт (0): 0
4 => 2-палубный
x = 1
y = 5
Вертикаль (1) или горизонт (0): 0
5 => 2-палубный
x = 9
y = 8
Вертикаль (1) или горизонт (0): 0
6 => 2-палубный
x = 10
y = 5
Вертикаль (1) или горизонт (0): 1
```

7 => 1-палубный
 x = 3
 y = 2
 Вертикаль (1) или горизонт (0): 0
 8 => 1-палубный
 x = 3
 y = 7
 Вертикаль (1) или горизонт (0): 0
 9 => 1-палубный
 x = 6
 y = 8
 Вертикаль (1) или горизонт (0): 0
 10 => 1-палубный
 x = 10
 y = 10
 Вертикаль (1) или горизонт (0): 0

Ваша карта:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | # | . | . | . | # | . | . | . |
| 3 | . | . | . | . | . | . | # | . | . | . |
| 4 | . | . | . | . | . | . | # | . | . | . |
| 5 | # | # | . | . | . | . | # | . | . | # |
| 6 | . | . | . | . | . | . | . | . | . | # |
| 7 | . | . | # | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | # | . | . | # | # |
| 9 | . | . | # | # | # | . | . | . | . | . |
| 10 | . | . | . | . | . | # | # | # | . | # |

Ваш ход!

Определите цель:

X = 7

Y = 3

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | # | . | . | . | # | . | . | . |
| 3 | . | . | . | . | . | . | # | . | . | . |
| 4 | . | . | . | . | . | . | # | . | . | . |
| 5 | # | # | . | . | . | . | # | . | . | # |
| 6 | . | . | . | . | . | . | . | . | . | # |
| 7 | . | . | # | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | # | . | . | # | X |
| 9 | . | . | # | # | # | . | . | . | . | . |
| 10 | . | . | . | . | . | # | # | # | . | # |

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | # | . | . | . | # | . | . | . |
| 3 | . | . | . | . | . | . | # | . | . | . |
| 4 | . | . | . | . | . | . | # | . | . | . |
| 5 | # | # | . | . | . | . | # | . | . | # |
| 6 | . | . | . | . | . | . | . | . | . | # |
| 7 | . | . | # | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | # | . | . | # | X |
| 9 | . | . | # | # | X | . | . | . | . | . |
| 10 | . | . | . | . | . | # | # | # | . | # |

Компьютер не попал!

Ваш ход!

Определите цель:

X = 7

Y = 5

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | 0 | . | . | . |
| 4 | . | . | . | . | . | . | . | . | . | . |
| 5 | . | . | . | . | . | . | X | . | . | . |
| 6 | . | . | . | . | . | . | . | . | . | . |
| 7 | . | . | . | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | . | . | . | . | . |
| 9 | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 8

Y = 5

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 7

Y = 4

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 7

Y = 6

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 6

Y = 5

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | 0 | . | . | . |
| 4 | . | . | . | . | . | . | 0 | . | . | . |
| 5 | . | . | . | . | . | X | X | 0 | . | . |
| 6 | . | . | . | . | . | . | 0 | . | . | . |
| 7 | . | . | . | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | . | . | . | . | . |
| 9 | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 5

Y = 5

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | 0 | . | . | . |
| 4 | . | . | . | . | . | . | 0 | . | . | . |

```

5 . . . . X X X 0 . .
6 . . . . . . 0 . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .
10 . . . . . . . . . .

```

Ваш ход!

Определите цель:

X = 4

Y = 5

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | 0 | . | . | . |
| 4 | . | . | . | . | . | . | 0 | . | . | . |
| 5 | . | . | . | X | X | X | X | 0 | . | . |
| 6 | . | . | . | . | . | . | 0 | . | . | . |
| 7 | . | . | . | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | . | . | . | . | . |
| 9 | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 10

Y = 3

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 9

Y = 3

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | 0 | . | X | 0 |
| 4 | . | . | . | . | . | . | 0 | . | . | . |
| 5 | . | . | . | X | X | X | X | 0 | . | . |
| 6 | . | . | . | . | . | . | 0 | . | . | . |
| 7 | . | . | . | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | . | . | . | . | . |
| 9 | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 9

Y = 2

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | . |
| 2 | . | . | . | . | . | . | . | . | X | . |
| 3 | . | . | . | . | . | . | 0 | . | X | 0 |
| 4 | . | . | . | . | . | . | 0 | . | . | . |
| 5 | . | . | . | X | X | X | X | 0 | . | . |
| 6 | . | . | . | . | . | . | 0 | . | . | . |
| 7 | . | . | . | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | . | . | . | . | . |
| 9 | . | . | . | . | . | . | . | . | . | . |

```

10 . . . . .
Ваш ход!
Определите цель:
X = 9
Y = 1
Точное попадание!
   1  2  3  4  5  6  7  8  9 10
1 . . . . . . . . X .
2 . . . . . . . . X .
3 . . . . . . 0 . X 0
4 . . . X X X X 0 . .
5 . . . X X X X 0 . .
6 . . . . . 0 . . .
7 . . . . . . . . .
8 . . . . . . . . .
9 . . . . . . . . .
10 . . . . . . . . .
Ваш ход!
Определите цель:
X = 10
Y = 5
Точное попадание!
   1  2  3  4  5  6  7  8  9 10
1 . . . . . . . . X .
2 . . . . . . . . X .
3 . . . . . 0 . X 0
4 . . . X X . 0 . .
5 . . . X X X X 0 . X
6 . . . . . 0 . . .
7 . . . . . . . . .
8 . . . . . . . . .
9 . . . . . . . . .
10 . . . . . . . . .
Ваш ход!
Определите цель:
X = 10
Y = 6
Точное попадание!
   1  2  3  4  5  6  7  8  9 10
1 . . . . . . . . X .
2 . . . . . . . . X .
3 . . . . . 0 . X 0
4 . . . . . 0 . . .
5 . . . X X X X 0 . X
6 . . . . . 0 . . X
7 . . . . . . . . .
8 . . . . . . . . .
9 . . . . . . . . .
10 . . . . . . . . .
Ваш ход!
Определите цель:
X = 10
Y = 7
Точное попадание!
   1  2  3  4  5  6  7  8  9 10
1 . . . . . . . . X .
2 . . . . . . . . X .
3 . . . . . 0 . X 0
4 . . . . . 0 . . .

```



```

5 . . . X X X X 0 . X
6 . . . . . . 0 . . X
7 . . . . . . . . . X
8 . . . . . . . . . .
9 . . . . . . . . . .
10 . . . . . . . . . .

```

Ваш ход!

Определите цель:

X = 6

Y = 8

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 7

Y = 7

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 8

Y = 7

Точное попадание!

```

      1  2  3  4  5  6  7  8  9 10
1 . . . . . . . . X .
2 . . . . . . . . X .
3 . . . . . . 0 . X 0
4 . . . . . . 0 . . .
5 . . . X X X X 0 . X
6 . . . . . . 0 . . X
7 . . . . . . 0 X . X
8 . . . . . 0 . . . .
9 . . . . . . . . . .
10 . . . . . . . . . .

```

Ваш ход!

Определите цель:

X = 8

Y = 8

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 6

Y = 7

Вы не попали. Играет компьютер

Компьютер попал!

```

      1  2  3  4  5  6  7  8  9 10
1 . . . . . . . . . 0
2 . 0 # . . . # 0 0 .
3 . . . . . . # . . .
4 . . 0 . . 0 # . . .
5 # # . . . . # . 0 #
6 . . . 0 . . . . . #
7 . . # . . . . . . .
8 . . . . . # . . # X
9 . . # # X . . . . .
10 . . . . . X # # . #

```

Компьютер не попал!

Ваш ход!

Определите цель:

X = 4

Y = 7

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 5

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 4

Y = 2

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 5

Y = 3

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 5

Y = 2

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | X | . |
| 2 | . | . | . | 0 | X | . | . | . | X | . |
| 3 | . | . | . | . | 0 | . | 0 | . | X | 0 |
| 4 | . | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | . | . | X | X | X | X | 0 | . | X |
| 6 | . | . | . | . | . | . | 0 | . | . | X |
| 7 | . | . | . | 0 | . | 0 | 0 | X | . | X |
| 8 | . | . | . | . | . | 0 | . | 0 | . | . |
| 9 | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 5

Y = 1

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | X | . | . | . | X | . |
| 2 | . | . | . | 0 | X | . | . | . | X | . |
| 3 | . | . | . | . | 0 | . | 0 | . | X | 0 |
| 4 | . | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | . | . | X | X | X | X | 0 | . | X |
| 6 | . | . | . | . | . | . | 0 | . | . | X |
| 7 | . | . | . | 0 | . | 0 | 0 | X | . | X |
| 8 | . | . | . | . | . | 0 | . | 0 | . | . |
| 9 | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 3

Y = 3

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | X | . | . | . | X | . |
| 2 | . | . | . | 0 | X | . | . | . | X | . |
| 3 | . | . | X | . | 0 | . | 0 | . | X | 0 |
| 4 | . | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | . | . | X | X | X | X | 0 | . | X |
| 6 | . | . | . | . | . | . | 0 | . | . | X |
| 7 | . | . | . | 0 | . | 0 | 0 | X | . | X |
| 8 | . | . | . | . | . | 0 | . | 0 | . | . |
| 9 | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 3

Y = 2

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | . | . | . | . | . | . | 0 |
| 2 | . | 0 | # | 0 | . | . | # | 0 | 0 | . |
| 3 | . | . | . | . | . | . | # | . | . | . |
| 4 | . | . | 0 | . | . | 0 | # | . | . | . |
| 5 | # | # | . | . | . | . | # | . | 0 | # |
| 6 | 0 | . | . | 0 | . | . | . | . | . | # |
| 7 | . | . | # | . | . | . | . | . | . | . |
| 8 | . | . | . | . | 0 | # | . | . | # | X |
| 9 | . | . | # | # | X | . | . | . | . | . |
| 10 | . | . | 0 | . | . | X | # | # | . | # |

Компьютер не попал!

Ваш ход!

Определите цель:

X = 2

Y = 3

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | 0 | . | . | . | . | . | 0 |
| 2 | . | 0 | # | 0 | . | . | # | 0 | 0 | . |
| 3 | . | . | . | . | . | . | # | . | . | . |
| 4 | . | . | 0 | . | . | 0 | # | . | . | . |
| 5 | # | # | . | . | . | . | # | . | 0 | # |
| 6 | 0 | . | . | 0 | . | . | . | . | . | # |
| 7 | . | . | # | . | . | . | . | . | . | . |
| 8 | . | . | . | . | 0 | # | . | . | # | X |
| 9 | . | . | # | # | X | . | . | . | . | . |
| 10 | . | . | 0 | . | . | X | # | # | . | # |

Компьютер не попал!

Ваш ход!

Определите цель:

X = 2

Y = 5

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | 0 | . | . | . | . | . | 0 |
| 2 | . | 0 | # | 0 | . | . | # | 0 | 0 | . |
| 3 | . | . | . | . | . | . | # | . | . | . |
| 4 | . | . | 0 | . | . | 0 | # | . | . | . |

```

5 # # . . . # 0 0 #
6 0 . . 0 . . . . #
7 . . # . . . . . .
8 . . . . 0 # . . # X
9 . . # # X . . . . .
10 . . 0 . . X # # . X

```

Компьютер не попал!

Ваш ход!

Определите цель:

X = 2

Y = 6

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 6

Вы не попали. Играет компьютер

Компьютер попал!

```

      1  2  3  4  5  6  7  8  9 10
1 . . . 0 . . . . . 0
2 . 0 # 0 . . # 0 0 .
3 . 0 . . . . # . . .
4 . . 0 . . 0 X . . .
5 # # . . . . # 0 0 #
6 0 . . 0 . . . 0 . #
7 . . # . . . . . . .
8 . . . . 0 # . . # X
9 . . # # X . . . . .
10 . . 0 . . X # # . X

```

Компьютер не попал!

Ваш ход!

Определите цель:

X = 2

Y = 9

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 8

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 7

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 2

Y = 7

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 4

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 7

Y = 1

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | . | 0 | . | . | . | . | . | 0 |
| 2 | . | 0 | # | 0 | 0 | . | # | 0 | 0 | . |
| 3 | . | 0 | . | . | . | . | # | . | . | . |
| 4 | . | . | 0 | . | . | 0 | X | . | 0 | . |
| 5 | # | # | . | . | . | . | # | 0 | 0 | # |
| 6 | 0 | . | . | 0 | . | . | . | 0 | . | # |
| 7 | . | 0 | # | . | . | . | . | . | . | . |
| 8 | . | . | . | . | 0 | # | . | . | # | X |
| 9 | . | . | # | # | X | . | . | . | . | . |
| 10 | . | . | 0 | . | . | X | # | # | . | X |

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 1

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 2

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 2

Y = 1

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 3

Y = 1

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | 0 | 0 | . | . | . | . | . | 0 |
| 2 | . | 0 | # | 0 | 0 | . | # | 0 | 0 | . |
| 3 | . | 0 | . | . | . | . | # | . | . | . |
| 4 | . | . | 0 | . | . | 0 | X | . | 0 | . |
| 5 | # | # | 0 | . | . | . | # | 0 | 0 | # |
| 6 | 0 | . | . | 0 | . | . | . | 0 | . | # |
| 7 | . | 0 | # | . | . | . | . | . | . | . |
| 8 | . | . | 0 | 0 | 0 | # | . | . | # | X |
| 9 | . | . | # | # | X | . | . | . | . | . |
| 10 | . | . | 0 | . | . | X | # | # | . | X |

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 3

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 5

Y = 7

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | 0 | 0 | . | X | . | 0 | . | X | . |
| 2 | 0 | . | 0 | 0 | X | . | . | . | X | . |
| 3 | 0 | 0 | X | . | 0 | . | 0 | . | X | 0 |
| 4 | 0 | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | 0 | . | X | X | X | X | 0 | . | X |
| 6 | 0 | 0 | . | . | . | . | 0 | . | . | X |
| 7 | 0 | 0 | . | 0 | X | 0 | 0 | X | . | X |
| 8 | 0 | . | . | . | . | 0 | . | 0 | . | . |
| 9 | . | 0 | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 5

Y = 8

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | . | . | 0 | 0 | . | . | . | . | . | 0 |
| 2 | . | 0 | # | 0 | 0 | . | # | 0 | 0 | . |
| 3 | . | 0 | . | . | . | . | # | . | . | . |
| 4 | . | . | 0 | . | . | 0 | X | . | 0 | . |
| 5 | # | # | 0 | . | . | . | # | 0 | 0 | X |
| 6 | 0 | . | . | 0 | . | . | . | 0 | . | # |
| 7 | . | 0 | # | . | 0 | . | . | . | 0 | . |
| 8 | . | . | 0 | 0 | 0 | # | . | . | # | X |
| 9 | . | . | # | # | X | . | . | . | . | . |
| 10 | . | . | 0 | . | . | X | # | # | . | X |

Компьютер не попал!

Ваш ход!

Определите цель:

X = 3

Y = 7

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 2

Y = 8

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 3

Y = 8

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 9

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | . | 0 | 0 | . | . | . | . | . | 0 |
| 2 | . | 0 | # | 0 | 0 | . | # | 0 | 0 | . |
| 3 | . | 0 | . | . | . | . | # | . | . | . |
| 4 | . | . | 0 | . | . | 0 | X | . | 0 | . |
| 5 | # | # | 0 | . | . | 0 | # | 0 | 0 | X |
| 6 | 0 | . | . | 0 | . | . | . | 0 | . | # |
| 7 | . | 0 | # | . | 0 | . | . | . | 0 | . |
| 8 | . | . | 0 | 0 | 0 | # | 0 | . | # | X |
| 9 | . | . | # | # | X | . | . | . | . | . |
| 10 | . | . | 0 | . | . | X | # | # | . | X |

Компьютер не попал!

Ваш ход!

Определите цель:

X = 1

Y = 10

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | 0 | 0 | . | X | . | 0 | . | X | . |
| 2 | 0 | . | 0 | 0 | X | . | . | . | X | . |
| 3 | 0 | 0 | X | . | 0 | . | 0 | . | X | 0 |
| 4 | 0 | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | 0 | . | X | X | X | X | 0 | . | X |
| 6 | 0 | 0 | . | . | . | . | 0 | . | . | X |
| 7 | 0 | 0 | 0 | 0 | X | 0 | 0 | X | . | X |
| 8 | 0 | 0 | 0 | . | 0 | 0 | . | 0 | . | . |
| 9 | 0 | 0 | . | . | . | . | . | . | . | . |
| 10 | X | . | . | . | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 2

Y = 10

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 3

Y = 9

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 3

Y = 10

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 4

Y = 9

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 4

Y = 10

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 5

Y = 9

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | 0 | 0 | . | X | . | 0 | . | X | . |
| 2 | 0 | . | 0 | 0 | X | . | . | . | X | . |
| 3 | 0 | 0 | X | . | 0 | . | 0 | . | X | 0 |
| 4 | 0 | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | 0 | . | X | X | X | X | 0 | . | X |
| 6 | 0 | 0 | . | . | . | . | 0 | . | . | X |
| 7 | 0 | 0 | 0 | 0 | X | 0 | 0 | X | . | X |
| 8 | 0 | 0 | 0 | . | 0 | 0 | . | 0 | . | . |
| 9 | 0 | 0 | 0 | 0 | X | . | . | . | . | . |
| 10 | X | 0 | 0 | 0 | . | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 5

Y = 10

Вы не попали. Играет компьютер

Компьютер не попал!

Ваш ход!

Определите цель:

X = 6

Y = 9

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | 0 | 0 | . | X | . | 0 | . | X | . |
| 2 | 0 | . | 0 | 0 | X | . | . | . | X | . |
| 3 | 0 | 0 | X | . | 0 | . | 0 | . | X | 0 |
| 4 | 0 | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | 0 | . | X | X | X | X | 0 | . | X |
| 6 | 0 | 0 | . | . | . | . | 0 | . | . | X |
| 7 | 0 | 0 | 0 | 0 | X | 0 | 0 | X | . | X |
| 8 | 0 | 0 | 0 | . | 0 | 0 | . | 0 | . | . |
| 9 | 0 | 0 | 0 | 0 | X | X | . | . | . | . |
| 10 | X | 0 | 0 | 0 | 0 | . | . | . | . | . |

Ваш ход!

Определите цель:

X = 8

Y = 9

Вы не попали. Играет компьютер

Компьютер попал!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | . | 0 | 0 | . | . | . | . | . | 0 |
| 2 | . | 0 | # | 0 | 0 | . | # | 0 | 0 | . |
| 3 | . | 0 | . | . | . | 0 | # | . | . | . |
| 4 | . | . | 0 | . | . | 0 | X | . | 0 | . |
| 5 | # | # | 0 | . | . | 0 | X | 0 | 0 | X |
| 6 | 0 | . | . | 0 | 0 | 0 | . | 0 | . | # |
| 7 | . | 0 | # | . | 0 | . | . | . | 0 | . |
| 8 | . | . | 0 | 0 | 0 | # | 0 | . | # | X |
| 9 | . | . | # | # | X | . | . | 0 | . | . |
| 10 | . | . | 0 | . | . | X | # | # | . | X |

Компьютер не попал!

Ваш ход!

Определите цель:

X = 8
Y = 10
Вы не попали. Играет компьютер
Компьютер не попал!
Ваш ход!

Определите цель:

X = 9

Y = 9

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | 0 | 0 | . | X | . | 0 | . | X | . |
| 2 | 0 | . | 0 | 0 | X | . | . | . | X | . |
| 3 | 0 | 0 | X | . | 0 | . | 0 | . | X | 0 |
| 4 | 0 | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | 0 | . | X | X | X | X | 0 | . | X |
| 6 | 0 | 0 | . | . | . | . | 0 | . | . | X |
| 7 | 0 | 0 | 0 | 0 | X | 0 | 0 | X | . | X |
| 8 | 0 | 0 | 0 | . | 0 | 0 | . | 0 | . | . |
| 9 | 0 | 0 | 0 | 0 | X | X | . | 0 | X | . |
| 10 | X | 0 | 0 | 0 | 0 | . | . | 0 | . | . |

Ваш ход!

Определите цель:

X = 10

Y = 9

Точное попадание!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | 0 | 0 | . | X | . | 0 | . | X | . |
| 2 | 0 | . | 0 | 0 | X | . | . | . | X | . |
| 3 | 0 | 0 | X | . | 0 | . | 0 | . | X | 0 |
| 4 | 0 | . | . | . | . | . | 0 | . | . | . |
| 5 | 0 | 0 | . | X | X | X | X | 0 | . | X |
| 6 | 0 | 0 | . | . | . | . | 0 | . | . | X |
| 7 | 0 | 0 | 0 | 0 | X | 0 | 0 | X | . | X |
| 8 | 0 | 0 | 0 | . | 0 | 0 | . | 0 | . | . |
| 9 | 0 | 0 | 0 | 0 | X | X | . | 0 | X | X |
| 10 | X | 0 | 0 | 0 | 0 | . | . | 0 | . | . |

Поздравляем, Вы победили!

Хотите повторить?

Да - 1

Нет - 0

=> => 1

Добро пожаловать в игру

"Морской бой"!

Вы должны расставить свои корабли.

Готовы?

Да - 1

Нет - 0

=> 0

Вы вышли!

Заключение

Для обучающихся составление программ компьютерных игр потребует большого интеллектуального напряжения и умений применять накопленные знания на практике. Желание решить сложную алгоритмическую задачу повлечет за собой углубление в мир языка программирования, разбор типовых задач и примеров реализации, чтение дополнительной литературы. Эксперименты, собственные находки и заготовки, работа в команде над большим проектом поможет ребятам эту задачу решать с удовольствием и получить удовлетворение от результатов своего труда, а также уверенности в правильном выборе будущей профессии.

Литература

1. Бриггс, Джейсон Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М. : Манн, 2016.
2. Васильев А. Н. Самоучитель С++ с примерами и задачами. 4-е издание
3. Дасгупта С. и др. Алгоритмы / С. Дасгупта, Х. Пападимитриу, У. Вазирани. М.: МЦНМО, 2014.
4. Доусон М. Изучаем С++ через программирование игр. - СПб.: Питер, 2016
Иванов и Фербер, 2017.
5. Конова Е. А., Поллак Г. А. Алгоритмы и программы. Язык С++: Учебное пособие. — 2-е изд., стер. — СПб.: Издательство «Лань», 2017. —384 с.
6. Костюк Ю. Л. Основы разработки алгоритмов : учебное пособие / Ю. Л. Костюк, И. Л. Фукс. - М.Ж БИНОМ. Лаборатория знаний, 2010. - 286 с.
7. Морской бой (игра). URL: [https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%80%D1%81%D0%BA%D0%BE%D0%B9_%D0%B1%D0%BE%D0%B9_\(%D0%B8%D0%B3%D1%80%D0%B0\)](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%80%D1%81%D0%BA%D0%BE%D0%B9_%D0%B1%D0%BE%D0%B9_(%D0%B8%D0%B3%D1%80%D0%B0))
8. Окулов С. М. Программирование в алгоритмах. 4-е изд.. М.: БИНОМ. Лаборатория знаний, 2013
9. Подготовка школьников к олимпиадам по информатике с использованием веб-сайта: учебно-методическое пособие для учащихся 7-11 классов. / А.В. Алексеев, С.Н. Беляев. – Ханты-Мансийск: РИО ИРО, 2008
10. Пэйн, Брайсон. Python для детей и родителей. М.: «Э», 2017
11. Роберт Седжвик. Алгоритмы на С++. Фундаментальные алгоритмы и структуры данных. — М.: «Вильямс», 2011. (ООП подход)
12. Скиена С. Алгоритмы. Руководство по разработке. СПб.: БХВ-Петербург, 2011
13. Скиена С. С., Ревилла М. А. Олимпиадные задачи по программированию. Руководство по подготовке к соревнованиям/Пер. с англ. – М: КУДИЦ-ОБРАЗ, 2005
14. У. Сэнд, К. Сэнд Hello World! Занимательное программирование. — СПб.: Питер, 2016. (Серия «Вы и ваш ребенок»)
15. Хайнеман Д., Поллис Г., Селков С. Алгоритмы. Справочник с примерами на С, С++, Java и Python. СПб.: ООО «Альфа-книга», 2017.
16. Space Travel. URL: https://ru.wikipedia.org/wiki/Space_Travel