Проект создан для защиты дипломной работы по курсам. Игра «Щелк» представляет собой следующее: двое по очереди "откусывают" от прямоугольной доски (размером 7 \* 5) кусочек. Игрок выбирает любое поле доски и снимает все фишки, которые находятся не ниже и не левее избранного поля (все фишки выше и правее). Снявший последнею левую нижнюю фишку - проигрывает.

**Про функцию предсказания хода.**

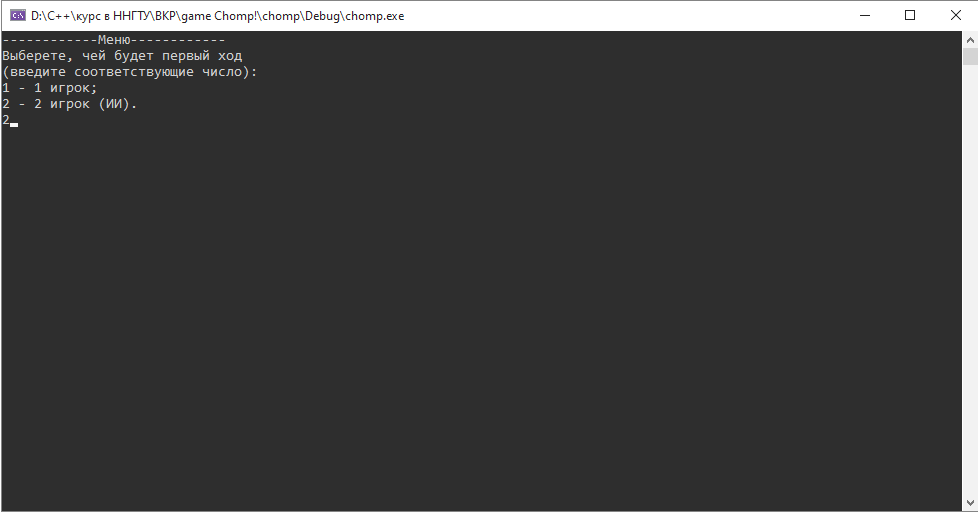
С точки зрения теории игр, «Щелк» является беспристрастной детерминированной игрой с полной информацией (т.е. игрокам известны правила игры, а также ходы других игроков). Наиболее известные выигрышные стратегии описываются для игры с размером поля 2хN, либо для квадратного поля.

Стратегию для игры позаимствовал у игры «Ним», в которой есть вариант игры с шоколадкой (т.е. прямоугольное поле с отравленным кусочком в углу). Данный вариант игры эквивалентен ниму с четырьмя кучками. Стратегия заключается в том, чтобы оставлять после своего хода позицию ним-сумм равной нулю. Ним-сумма является результатом сложения всех размеров кучек в двоичной системе счисления без учета переноса разрядов. Для этого поле игры разделяем на четыре кучки, которые будут соответствовать строкам нашей прямоугольной области (так как у нас пять строк, первую строчку в расчет не берем, потому что на конечную цель игры для игрока она влияет не сильно). В каждой строке идет подсчет свободных фишек и перевод их в двоичную систему исчисления. После считается общая сумма всех четырех кучек поразрядно без учета переноса разряда.

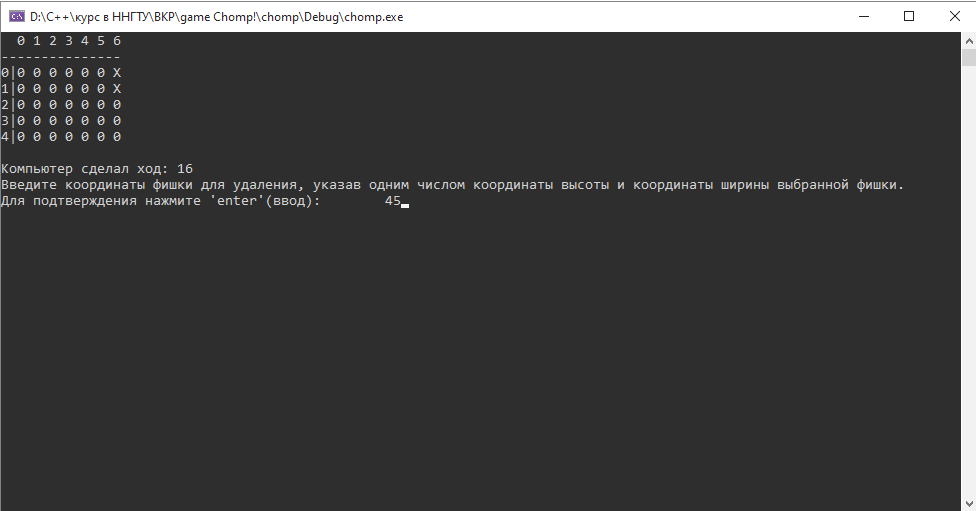
Подведем итог: компьютер делает ход (выбирает фишку для ее поглощения) таким образом, чтобы после его хода ним-сумма была равна нулю (при этом, как минимум одну фишку он должен съесть). Также в подсчете последней кучи (последней строки области игры с «отравленной» долькой) не учитывается «отравленная» долька, потому что при некоторых сценариях игры компьютеру пришлось бы выбирать для поглощения эту дольку для сохранения правила ним-сумм.

**Пример игры с ботом.**

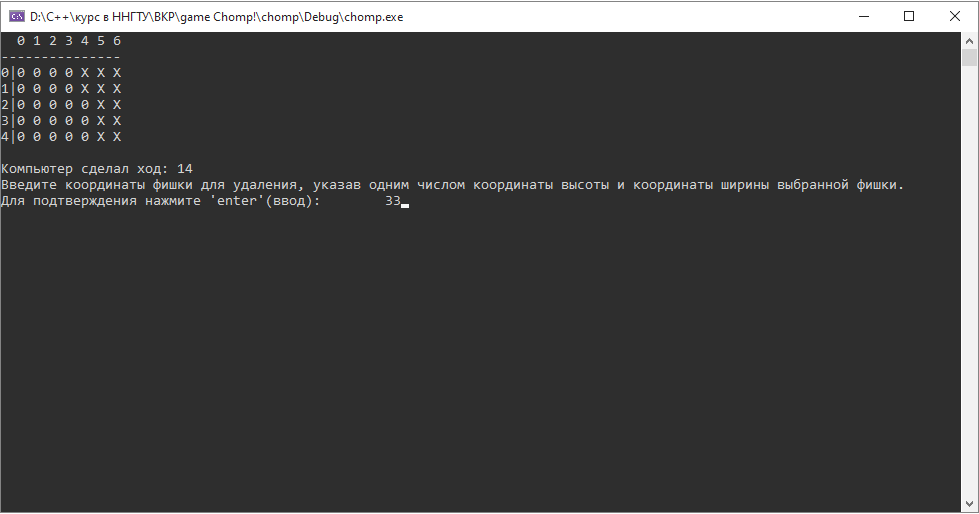
В меню игры выбирает игрока, который делает ход первым. В данной игре первый ход отдадим компьютеру. Отравленная фишка находится на координатах 40 (где первое число – цифра соответствующая строке, вторая цифра – соответствует столбцу).



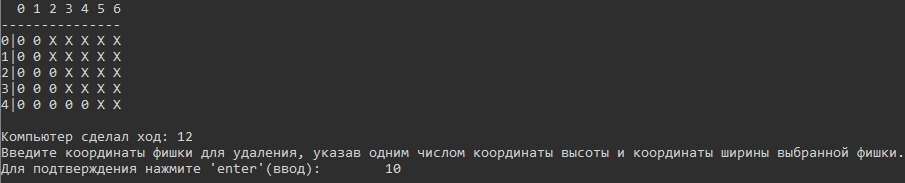
После хода компьютера (выбрал фишку по координатам 16) делаем ход фишкой 45.



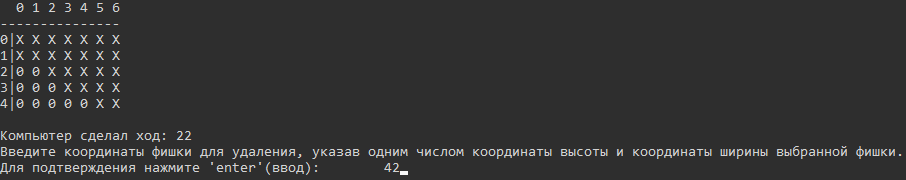
Далее компьютер сделал ход фишкой 14, а мы выбираем фишку 33.



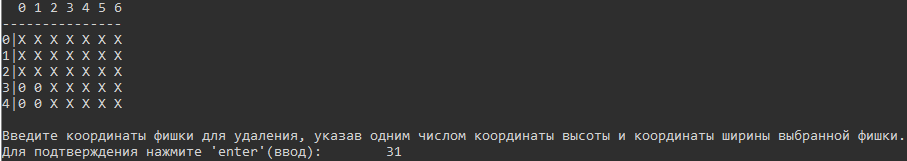
Компьютер съедает фишку 12, мы делаем ход фишкой 10.



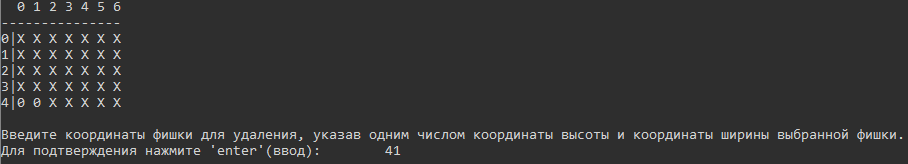
После компьютер выбирает фишку 22, мы – 42.



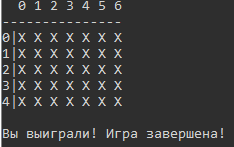
Компьютер съедает фишку 20, мы выбираем фишку 31.



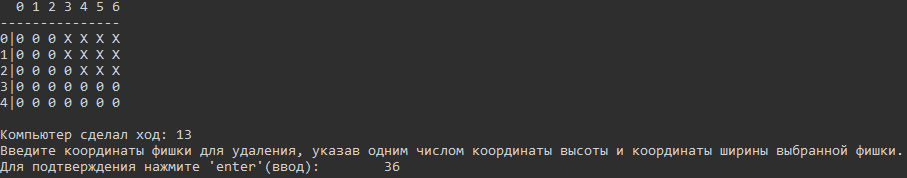
Компьютер выбирает фишку 31, мы – 41. Таким ходом, мы не оставляем компьютеру выбора, кроме как съесть отравленную часть.



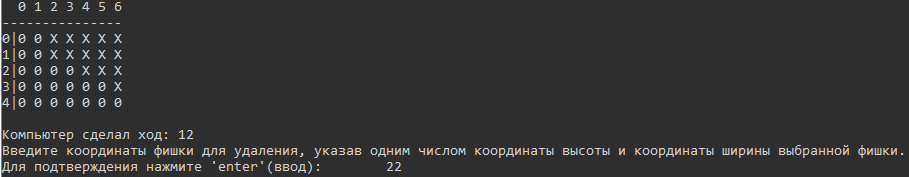
В итоге, в игре с компьютером нам удалось выиграть.



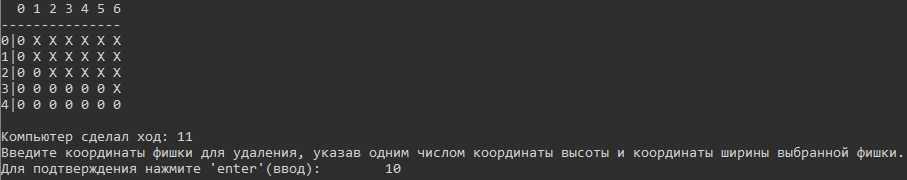
Сыграем еще одну партию, проверив как играет компьютер в других условиях. В этот раз первый ход будет за нами. Мы выбираем фишку 24. Компьютер делает ход фишкой 13, мы отвечаем фишкой 36.



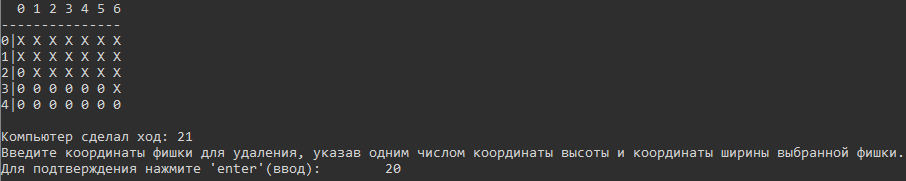
Далее программа выбирает фишку 12, мы – 22.



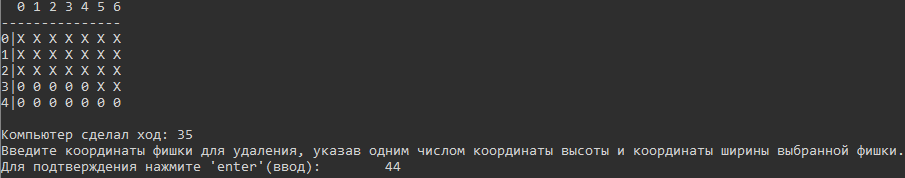
После компьютер выбирает фишку 11, мы съедаем фишку 10.



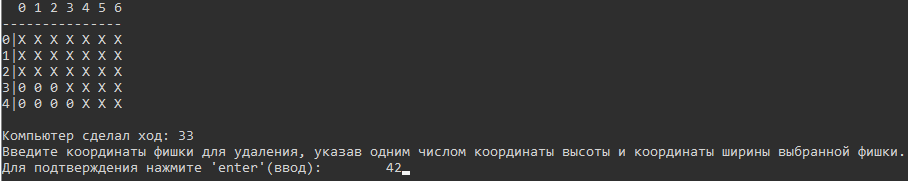
Компьютер выбирает фишку 21, мы – 20.



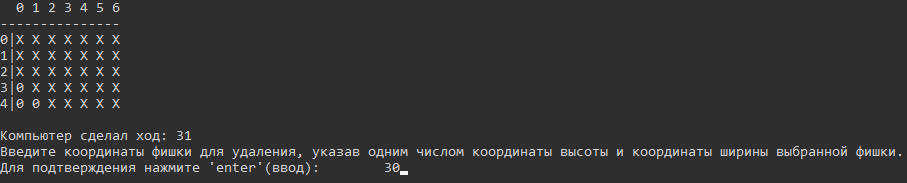
Программа съедает позицию 35, мы выбираем 44.



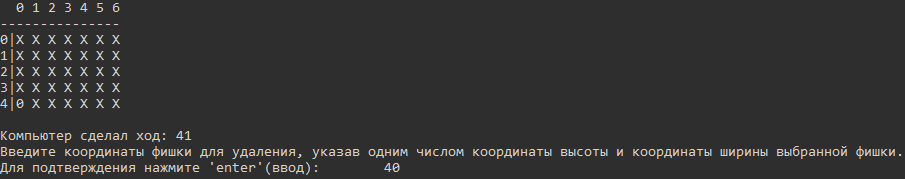
Компьютер выбирает координаты 33, мы – 42.

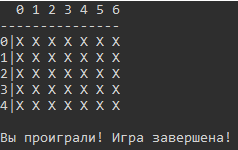


Далее компьютер ходит фишкой 31, а мы фишкой 30.



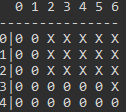
В конце, компьютер съедает последнюю неотравленную фишку 41, а нам остается только 40.





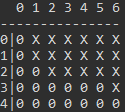
**Пример расчета хода компьютера**:

Для примера расчёта хода компьютера возьмем реальное расположение фишек, которые были в одной из игре. Их расположение представлено на картинке ниже.



Программа начинает посчитывать количество свободных фишек в каждой куче (одна куча является одной строкой массива, кроме самой первой с индексом 0). В итоге компьютер рассчитывает, что в первой куче у него две свободных фишек, во второй – тоже две, в третьей – шесть, в четвертой – тоже шесть (потому что отравленную фишку в подсчет не берем). Так как компьютер не может не сделать ход, программа начинает убирать по одной фишке, начиная с самой высокой кучи (с первой кучи). В случае если в рассматриваемой кучи нет свободных фишек (или при их вычитании сумма ним-сумма не равна 0) – переходим к следующей кучи, пока программа не найдет позицию свободных фишек, при которой ним-сумма будет равна 0. В данном случае, компьютеру не пришлось много считать, и он нашел позицию ним-сумму равной нулю, при которой в первой куче будет одна фишка, во второй – две, третьей и четвертой по шесть. Получается, программе нужно сделать такой ход, по итогу которого количество свободных фишек будет равно посчитанной ранее самой программой. В конечном счете, программа убирает одну фишку из первой кучи для сохранения ним-суммы равной 0.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| номер кучи | Кол-во СФ | Первый разряд | Второй разряд | Третий разряд |
| 1 | 1 | 0 | 0 | 1 |
| 2 | 2 | 0 | 1 | 0 |
| 3 | 6 | 1 | 1 | 0 |
| 4 | 6 | 1 | 1 | 0 |
| Сумма куч по разрядам | | 0 | 1 | 1 |
| Итоговая сумма (первого, второго и третьего разрядов) равна 0 | | | | |



**Примеры, если мы хотим сходить не по правилам (защита от дурака).**

Все наши действия с пользователем происходит через командную строку. От игрока мы хотим получить число, которое будет соответствовать координатам фишки для удаления, где первая цифра – это номер строки, а вторая – номер столбца. Из этого следует, что в случае, если игрок вводит не целое число, вводит слишком длинное число или вводит символы – появляется надпись о не корректном значении. В случае, если фишка уже занята – также выводим об этом информацию пользователю.

