УДК 004

Различные подходы к решению конечных игр

Бабошина Анна Викторовна
Ярославский государственный педагогический университет им. К.Д.Ушинского
Студент

Корнилов Петр Анатольевич
Ярославский государственный педагогический университет им. К.Д.Ушинского
к.ф.-м.н., доцент, доцент кафедры теории и методики обучения информатике

Аннотация
Была разработана программа, визуализирующая различные алгоритмы решения парных конечных игр. В работе рассматриваются алгоритм мини-макс, числа Спрага-Грюнди и использование двоичной системы счисления для решения таких конечных математических игр как «Определитель», «24 карты» и «Ним». В каждой игре присутствуют три режима, в двух из которых («Человек - Компьютер», «Компьютер - Человек») пользователю наглядно представлен алгоритм игры компьютера, написанный в соответствии с выигрышной стратегией. Помимо этого, каждая игра включает в себя правила игры и справочный материал.
Ключевые слова: конечные игры, цена игры, числа Спрага-Грюнди, система счисления.

Different approaches to the solution of finite games

Baboshina Anna Victorovna
Yaroslavl state pedagogical university after K.D.Ushinsky
Student

Kornilov Petr Anatolievich
Yaroslavl state pedagogical university after K.D.Ushinsky
candidate of physico-mathematical sciences, associate Professor, associate Professor of theory and methodology of teaching Informatics

Abstract
The program was created that visualizes various algorithms for solving paired finite games. The paper considers the mini-max algorithm, Sprague-Grundy numbers and the use of the binary number system for solving such finite mathematical games as « Determinant», «24 cards» and «Nim». In each game there are three modes, in two of which («Human - Computer», «Computer -

Human») algorithm of computer game is demonstrated to user, written in accordance with the winning strategy. In addition, each game includes game rules and reference material.

Keywords: the finite game, the price of the game, Sprague-Grundy numbers, binary system.

Введение

В современный век компьютерных технологий трудно представить себе жизнь человека без использования компьютера. Компьютеры настолько плотно вошли в нашу жизнь, что их используют не только взрослые, но и юноши, подростки и даже дети. Среди всех ресурсных возможностей компьютера и интернет пространства особое место занимают игры (особенно среди подростков и юношей). В связи с этим многие исследователи отмечают снижение у учащихся мотивации к учебной деятельности. Таким образом, в настоящее время становятся всё более актуальными вопросы мотивации учащихся. Одним из возможных способов решения проблемы является привлечение учащегося в учебную деятельность посредством использования привычной для него среды. Мы считаем, что использование логических игр, представленных с помощью интерактивной среды, повысит интерес учащихся при изучении математики и информатики и, в частности, различных алгоритмов. Кроме того, наличие подобной интерактивной среды позволит углубить уже имеющиеся знания и навыки разработки алгоритмов.

Таким образом, целью нашей работы стала разработка программы, содержащей набор математических игр. На основе данной программы в игровой форме изучаются некоторые алгоритмы для решения конечных математических игр.

В соответствии с вышеизложенной целью были поставлены следующие задачи:

1. Проанализировать и подобрать соответствующие конечные математические игры.
2. Реализовать алгоритмы выбранных игр.
3. Визуализировать алгоритмы.
4. Сделать удобный и доступный для пользователя интерфейс.

1. Теория игр

Игра – упрощенная формализованная модель конфликтной ситуации. Игрок – одна из сторон в игровой ситуации. Каждый игрок может иметь свои стратегии. Стратегией игрока называется одно из возможных решений из множества допустимых решений этого игрока. Каждый из участников игры может выбирать свою стратегию. Совокупность стратегий, которые выбрали участники игры, называется игровой ситуацией [5]. Игра называется конечной, если у каждого игрока конечное число стратегий. Игра называется парной, если в ней участвуют два игрока.

2. Игра «Определитель»

2.1.Правила игры

Играют двое. Игроки А и В. Они по очереди вписывают в таблицу 3 х 3 числа от 1 до 9 (каждое число должно быть использовано). Когда таблица заполнена, подсчитывают две суммы: сумму SA произведений по строкам и сумму SB произведений по столбцам. Если SA > SB, то выигрывает игрок А; если SA < SB, то выигрывает игрок В.

2.2.Терминология

Оценить игровую ситуацию можно определив цену игры, которая в игре «Определитель» является разницей суммы произведений по строкам и суммы произведений по столбцам. Для первого игрока, который стремится набрать максимум из суммы произведений по строкам, эта разность должна быть максимальной, для второго же, наоборот – минимальной.

Для решения данной игры на компьютере строится дерево игровых ситуаций в соответствии с методом мини-макс, то есть в соответствии с критерием Вальда [4]. Критерий Вальда (максиминный критерий) — один из критериев принятия решений в условиях неопределённости. По критерию Вальда за оптимальную принимается стратегия, которая в наихудших условиях гарантирует максимальный выигрыш.

2.3.Описание алгоритма

Игра «Определитель» на первый взгляд не кажется сложной. Игроки делают всего девять ходов – просчитаем все варианты и выберем нужный. Но различных партий очень много. Для первого хода есть 81 вариант (одно из девяти чисел в одну из девяти ячеек), для второго хода 64 варианта и т. д. Всего получается $(9!)^2$ = 131681894400 различных партий. Определить результат игры таким перебором невозможно. В связи с этим в режимах игры «Человек - компьютер» и «Компьютер - Человек» компьютер делает ходы в соответствии с методом мини-макс.

Как было сказано выше, для решения данной игры на компьютере строится дерево игровых ситуаций в соответствии с методом мини-макс. В программе алгоритм реализован с помощью рекурсивной процедуры, основанной на максиминном критерии.

Алгоритм заключается в рассмотрении всех возможных исходов, опирающихся на цену игры, которая должна быть максимальна для первого игрока и минимальна для второго. В связи с чем, делая, например, нечетный ход – ход первого игрока, компьютер выбирает максимум из минимумов следующего четного хода противника. Минимум выбирается из максимумов следующего нечетного и т.д.

В процедуре прописаны действия для каждого из ходов, за исключением первых двух. Что касается первого хода, он был просчитан с помощью режима игры двух компьютеров, при котором была вычислена максимальная цена игры из возможных девяти вариантов. Второй ход в программе прописан вручную на основе вычислений алгоритма. Это связано со временем работы программы при вычислении всех исходов для второго хода (время работы около 10 минут).

На рис. 1 приведен фрагмент игры в режиме «Человек – компьютер». После седьмого хода, который сделал человек, в доступе компьютера осталось два числа – «3» и «4». На рисунке мы можем увидеть, какой вариант восьмого хода выбрал компьютер.

Почему компьютер сходил именно так? Поскольку задача машины набрать максимум из суммы произведений по столбцам, на данном ходе производится поиск минимальной цены игры. Возможны два варианта: цена игры будет равна

-79 = (9*1*7 + 4*5*3 + 6*8*2) – (9*4*6 + 1*5*8 + 7*3*2), или же

-39 = (9*1*7 + 3*5*4 + 6*8*2) – (9*3*6 + 1*5*8 + 7*4*2). Выбор компьютера очевиден.

Рисунок 1

Пояснение к названию «Определитель»

Если мы представим заполненную таблицу в виде матрицы,

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix}$$

то цена игры:

$$a_1 * a_2 * a_3 + a_4 * a_5 * a_6 + a_7 * a_8 * a_9$$
$$- a_1 * a_4 * a_7 - a_2 * a_5 * a_8 - a_3 * a_6 * a_9 \cdot$$

Что является расчетом определителя матрицы вида:

$$\begin{pmatrix} a_1 & a_6 & a_8 \\ a_9 & a_2 & a_4 \\ a_5 & a_7 & a_3 \end{pmatrix}.$$

2.4.Визуализация алгоритма с помощью деревьев

Как было сказано ранее, в программе наглядно представлено частичное построение деревьев игровых ситуаций. Деревья строятся только для седьмого и восьмого ходов, так как у дерева игры, например, шестого хода уже 288 веток. Для начала рассмотрим построение дерева игры для уже разобранного нами примера.

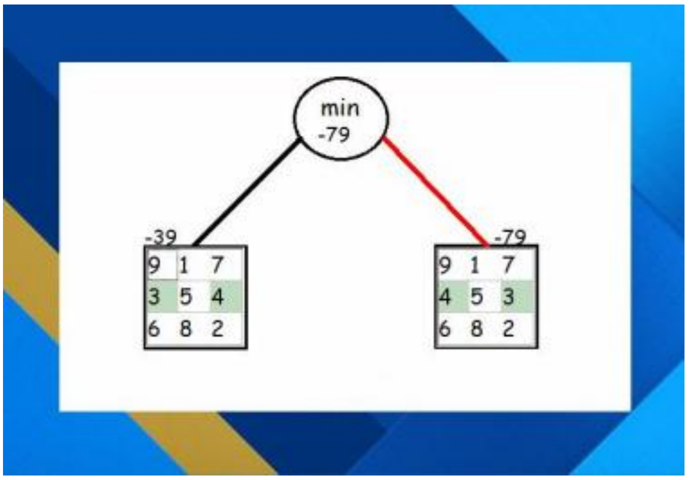

Рисунок 2

На рисунке 2 представлено дерево игры для восьмого хода. Мы уже знаем, что на данном ходе производится поиск минимальной цены игры. Нужная нам ветка дерева обозначена красным цветом, зеленым цветом выделены незаполненные до восьмого хода ячейки. Также в обоих вариантах указана цена игры.

Теперь рассмотрим выбор седьмого хода. На рисунке 3 приведен фрагмент игры в режиме «Компьютер – человек». После шестого хода, который сделал человек, в доступе компьютера осталось три числа – «6», «7» и «8». На рисунке мы можем увидеть, какой вариант седьмого хода выбрал компьютер.

Рисунок 3

В таком случае задача машины - набрать максимум из суммы произведений по строкам, на данном ходе производится поиск максимальной цены игры. Возможны девять вариантов. На рис. 4 рассматривается дерево игровых ситуаций для данного примера. Здесь наиболее наглядно представлен метод мини-макс (максиминный критерий).

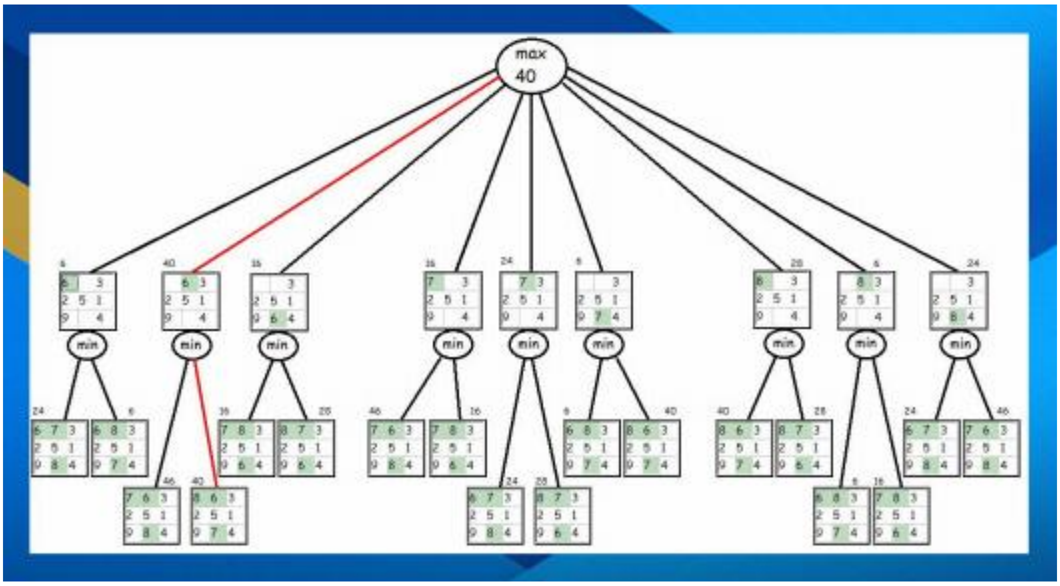

Рисунок 4

3. Игра «24 карты»

3.1.Правила игры

Играют двое. На столе 24 карты достоинством от 1 до 6 (масть не имеет значения). Каждый игрок на своем ходе может взять со стола любую карту. Карта суммируется с общим количеством взятых со стола карт. Победит тот, кто на своем ходе наберет ровно 50, или заставит противника набрать > 50 очков.

3.2.Терминология

В игре «24 карты» позиция может быть полностью охарактеризована числом и достоинством карт, оставшихся на столе. Выигрывающая позиция соответствует сумме, равной пятидесяти из карт, отсутствующих на столе. Спраг и Грюнди предложили (соответственно в 1936 и 1939 годах) связывать с каждой игровой позицией неотрицательное целое число следующим образом:

- выигрывающей позиции сопоставляется ноль;

- данной игровой позиции сопоставляется наименьшее неотрицательное целое, отличающееся от чисел, связанных с позициями, которые могут быть достигнуты за один ход, исходя из данной [3].

Из позиции ноль существуют ходы только в ненулевую позицию. Из ненулевой позиции существует хотя бы один ход в позицию ноль.

3.3.Описание алгоритма

Исходя из пункта 3.2. необходимо определить все возможные позиции. Имея карты шести разных достоинств, заведем шестимерный массив. Первый индекс массива будет соответствовать количеству карт достоинством 1, второй индекс количеству карт достоинством 2, третий количеству карт достоинством 3 и т.д.

Изначально определим значения элементов массива для позиций, в которых количество набранных очков равно 50. Эти позиции являются выигрывающими, следовательно, значения элементов массива для этих позиций можно сразу присвоить нулю. Позиции, в которых количество набранных очков >50 являются проигрышными, поэтому для этих позиций можно сразу определить целое положительное число (например, 10). Дальнейшие вычисления основываются на уже определенных нами позициях. Заполнение массива производится в соответствии с количеством набранных очков, причём по убыванию (49 очков, 48, 47 и т.д.).

Рассмотрим пример определения значения элемента массива, количество очков которого равно 49 (SG[4, 3, 2, 2, 1, 1]). Вспомним, что данной игровой позиции сопоставляется наименьшее неотрицательное целое, отличающееся от чисел, связанных с позициями, которые могут быть достигнуты за один ход, исходя из данной. А именно:

- SG[3, 3, 2, 2, 1, 1] = 0 (сумма очков = 50);
- SG[4, 2, 2, 2, 1, 1] = 10 (сумма очков = 51);
- SG[4, 3, 1, 2, 1, 1] = 10 (сумма очков = 52) и т.д.

Из расчетов видно, что значение элемента массива SG[4, 3, 2, 2, 1, 1] = 1 (наименьшее целое неотрицательное число отличное от 0 и 10).

Далее компьютер делает ходы в соответствии с заполненным массивом.

3.4.Визуализация алгоритма

На рис. 5 приведен фрагмент игры в режиме «Человек – компьютер». В нижней части рисунка зеленым цветом выделен элемент массива, позиция, в которой оказался компьютер (SG[2, 3, 4, 2, 4, 4] = 3). И также шесть позиций, которые могут быть достигнуты из данной. Задача компьютера, находясь в ненулевой ячейке, выбрать нулевую.

Таким образом, компьютер берет карту достоинством 4, что приводит его в элемент SG[2, 3, 4, 1, 4, 4] = 0. Этот элемент выделен на рисунке красным цветом.

Но как должен поступить компьютер, если он оказался в нулевой позиции? В этом случае лучшего хода не существует, компьютер находится в проигрышной позиции, поэтому ход выбирается рандомно.

На рис. 6 приведен фрагмент игры, в котором компьютер оказывается в нулевой позиции (SG[2, 3, 4, 3, 3, 4] = 0). Желтым цветом выделен элемент массива, который компьютер выбирает рандомно.

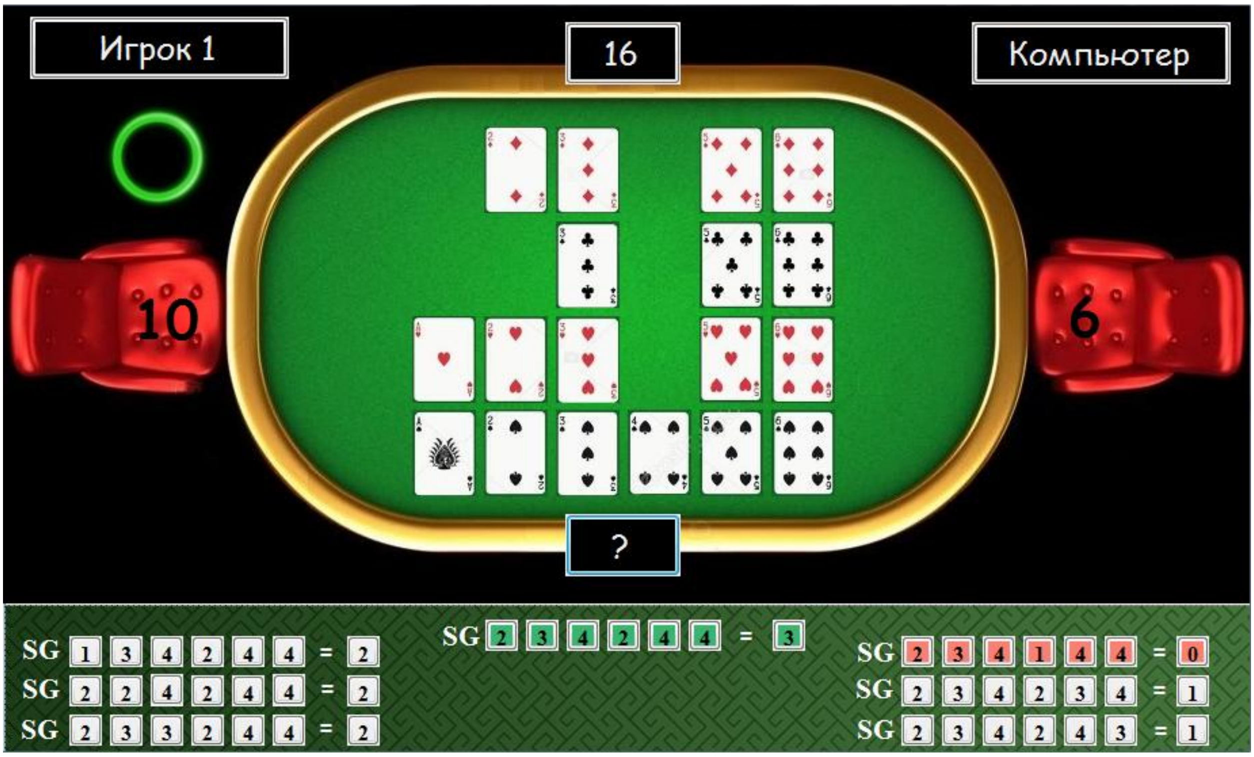

Рисунок 5

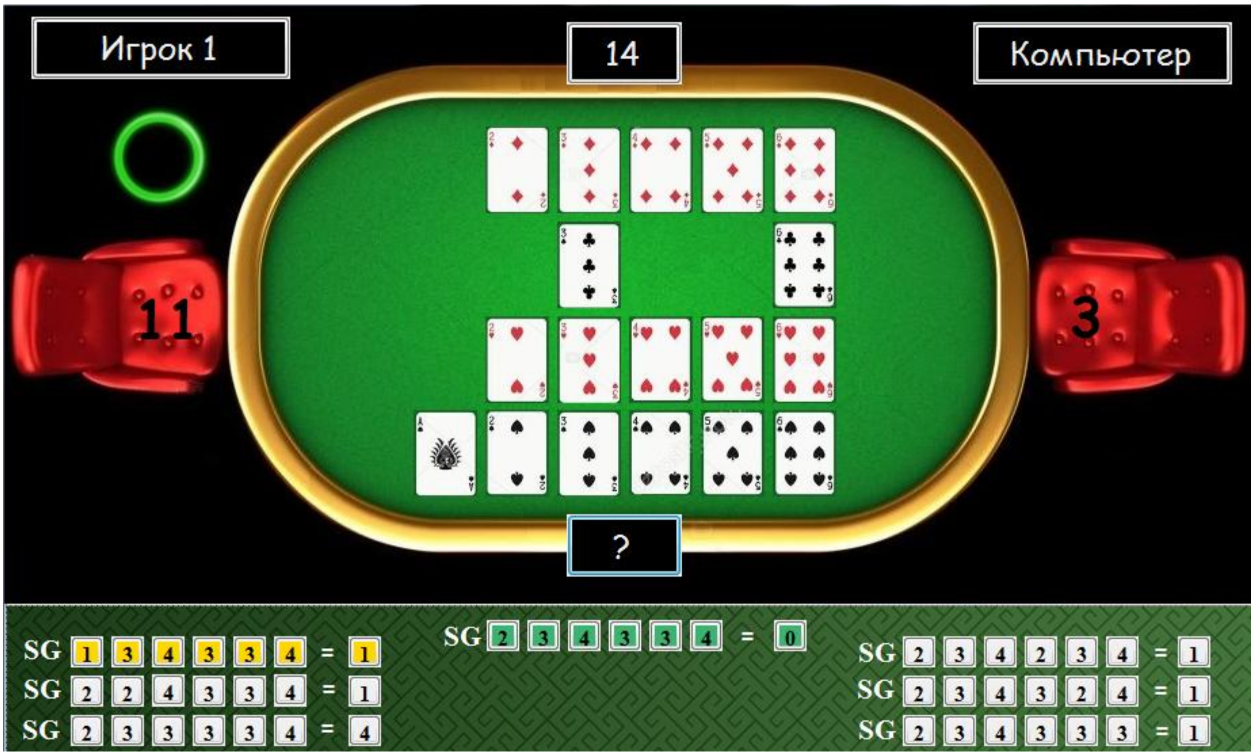

Рисунок 6

Исходя из вышесказанного, можно сделать вывод, что если одному из игроков удалось отправить соперника в нулевую позицию, то при безошибочной игре, можно одержать победу.

4. Игра «Ним»

4.1.Правила игры

Игра для двоих. На столе пять кучек, и в каждой куче – некоторое количество камней. Каждый из игроков на своем ходе берет столько камней, сколько хочет, но только из одной кучки и обязательно хотя бы один. Выиграет тот, кто берет последний камень из последней кучки.

4.2.Терминология

В игре «Ним» каждую комбинацию камней можно назвать либо «опасной», либо «безопасной». Если позиция, создавшаяся после очередного хода игрока, гарантирует ему выигрыш, она называется безопасной; в противном случае позиция называется опасной. Любую опасную позицию, сделав соответствующий ход, всегда можно превратить в безопасную. Каждая безопасная позиция становится опасной после любого хода [1].

4.3.Описание алгоритма

Чтобы определить, опасна или безопасна данная позиция, число камней в каждом ряду нужно записать в двоичной системе. Если сумма чисел в каждом столбце (разряде) равна нулю или четна, то позиция безопасна, если же сумма нечетна хотя бы в одном разряде, то позиция опасна.

Рациональная игра заключается в том, чтобы каждый раз превращать опасную позицию в безопасную. Рассмотрим, как это сделать с помощью двоичной системы счисления [1].

4.4.Визуализация алгоритма

На рис. 7 приведен фрагмент игры в режиме «Человек - Компьютер». Мы видим 5 рядов камней: 1 камень в первом ряду, 6 во втором, 6 в третьем, 2 в четвертом и 3 в пятом. Именно такая позиция сложилась после хода компьютера. Справа визуализирован алгоритм действий компьютера по превращению опасной позиции в безопасную.

Изначально в пятом ряду было 9 камней. После записи количества камней в каждом ряду в двоичной системе, был найден самый левый столбец с нечетной суммой цифр (первый столбец). Изменив любой ряд с единицей в этом столбце, можно превратить позицию в безопасную. В данном случае такой ряд всего один и соответствует числу 9.

Проделаем над числом 9 такую операцию: те цифры двоичной записи этого числа, под которым оказались крестики, заменим на противоположные (0 на 1 и 1 на 0), остальные цифры оставим без изменения [2].

Получилась двоичная запись числа 3. Компьютер совершает ход, оставив в пятом ряду из 9 камней 3, забрав 6.

Если же компьютер окажется в безопасной позиции, то ход совершается рандомно, так как каждая безопасная позиция становится опасной после любого хода. Как, например, на рис. 8, находясь в опасной позиции, компьютер берет 1 камень из пятого ряда.

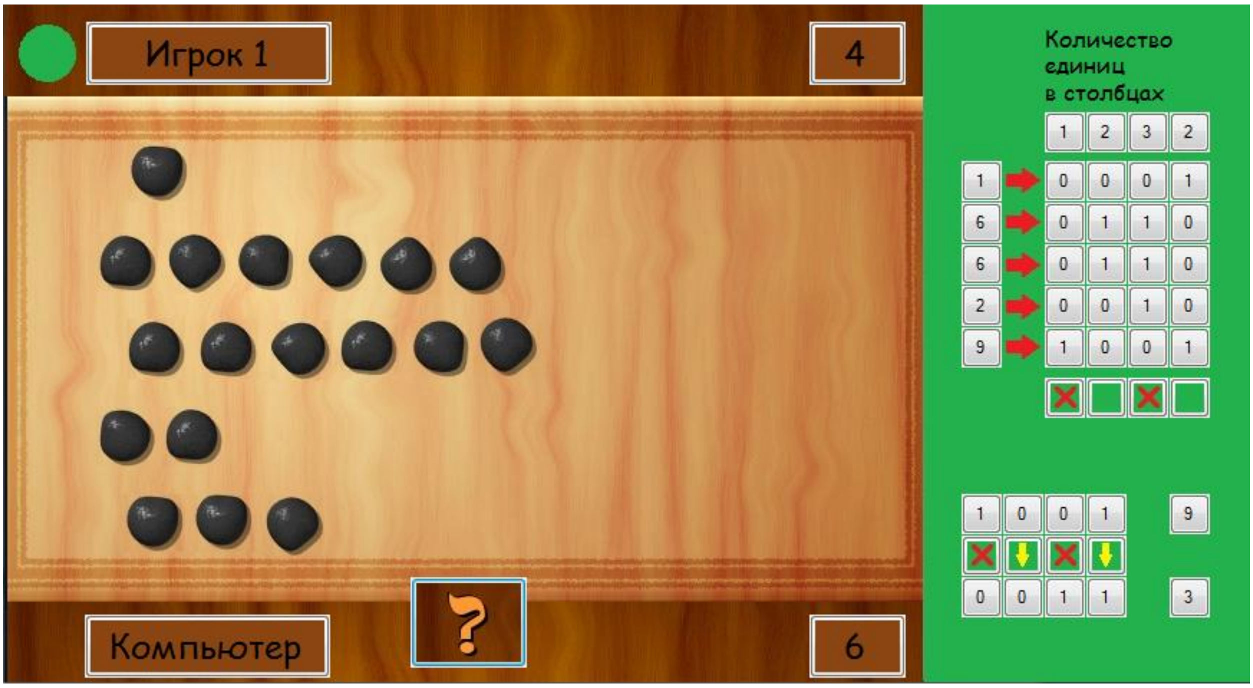

Рисунок 7

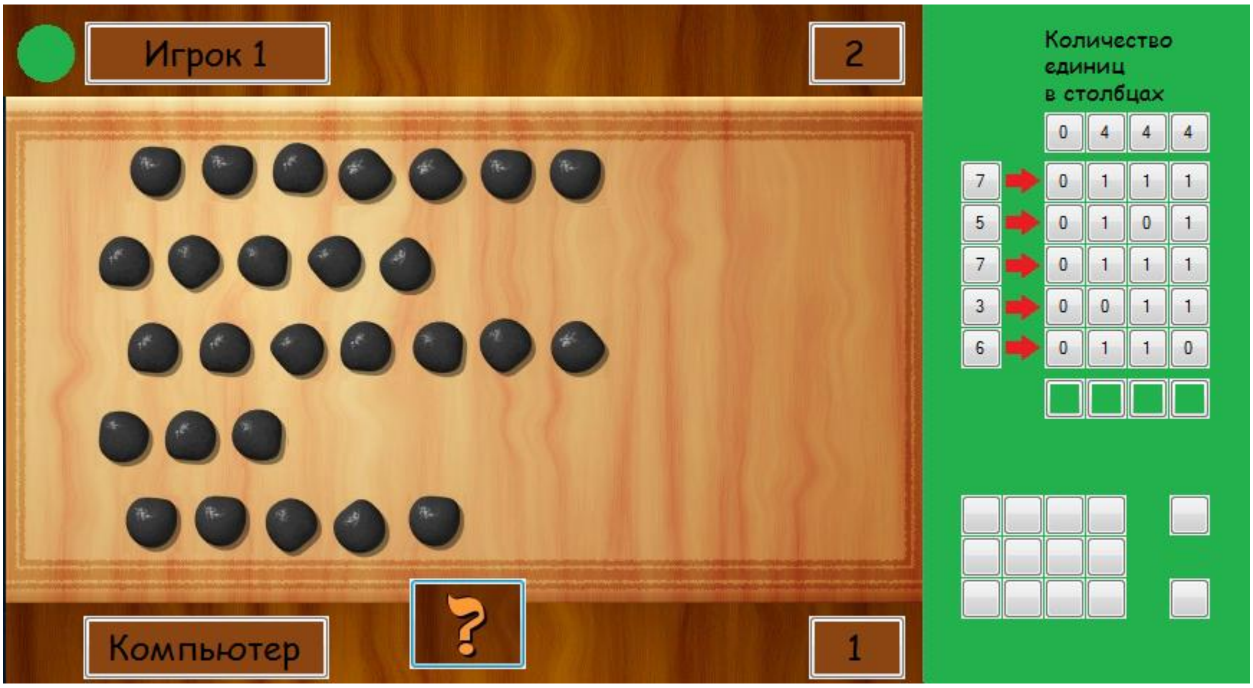

Рисунок 8

5. Заключение

В данной программе осуществлен подбор конечных математических игр, алгоритмы реализованы и наглядно представлены. Программа обладает удобным и доступным для пользователя интерфейсом.

Задачи решены, цель достигнута.

В дальнейшем планируется расширить набор математических игр и добавить обучающий режим для каждой игры. Программы подобраны таким образом, что у учащихся будет возможность написать свою процедуру, в которой будет реализован подходящий на их взгляд алгоритм. В связи с чем, в программе будут добавлены режимы «Битвы процедур».

Библиографический список

1. Гарднер М. Математические головоломки и развлечения. М.: АСТ; Зебра Е, 2010. 640 с.
2. Пискарев А.В. Игра «Ним». Оценка игровой ситуации. Алгоритм игры // Компьютерные инструменты в образовании. 2001. №1. С.87-89.
3. Арсак Ж. Программирование игр и головоломок: М.: Наука. Гл. ред. физ.-мат. лит., 1990. 224 с.
4. https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D1%82%D0%B5%D1%80%D0%B8%D0%B9_%D0%92%D0%B0%D0%BB%D1%8C%D0%B4%D0%B0
5. https://math.semestr.ru/games/maingames.php