

ВСЕРОССИЙСКАЯ КОНФЕРЕНЦИЯ, 2014 ГОД

Исследования и практика - путь к новым знаниям

Никаноров Александр Андреевич

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский университет «Высшая школа экономики»

Город Москва

ИГРА «КАКУРО» НА ПЛАТФОРМЕ IOS

В статье рассматривается разработка программной реализации игры «Какуро» на платформе iOS. В программе случайно генерируется игровое поле, соответствующее правилам игры Какуро, пользователь интерактивно решает сгенерированную головоломку.

Разработаны алгоритмы генерации клеток на игровом поле в соответствии с правилами Какуро, нахождения неповторяющихся слагаемых числа, использованы основные принципы создания пользовательского интерфейса для приложений на платформе IOS 6. Кроме того, в приложении реализована функция сохранения и загрузки состояния игр, функция интерактивного создания пользовательского игрового поля и режим обучения.

От своих аналогов приложение отличается обширным набором функций, таких как сохранение состояний до трех игр одновременно и интерактивное создание пользовательского игрового поля.

Ключевые слова и фразы: Какуро, игра Какуро платформа IOS

Научный руководитель:

*Доцент кафедры УРПО отделения Программной инженерии Национального исследовательского университета «Высшая школа экономики»
Ахметсафина Римма Закиевна*



Введение

Какуро (Cross Sums) – числовая головоломка, которую также часто называют математическим аналогом кроссворда. Размер поля головоломки обычно 16x16, но может и варьироваться. Поле состоит из клеток трех типов: цифровые (рис. 1а), клетки с подсказками (рис. 1б) и пустые клетки (рис. 1с).

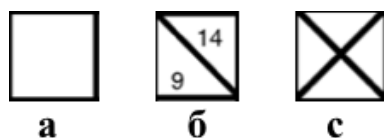


Рис. 1

Цифровые клетки – это клетки, которые в результате игры должны быть заполнены цифрами.

Пустые клетки – это клетки, которые являются ограничителями, в них нет и не может быть цифр.

Клетки с подсказками разделены диагональю слева направо сверху вниз. В правой верхней части клетки записано число, которое должно получиться при сложении значений всех цифровых клеток по горизонтали правее текущей до пустой клетки или клетки с подсказкой (рис. 2а). Число в левой нижней части клетки должно получиться при сложении значений всех цифровых клеток по вертикали ниже текущей и до следующей пустой клетки или клетки с подсказкой (рис. 2б).

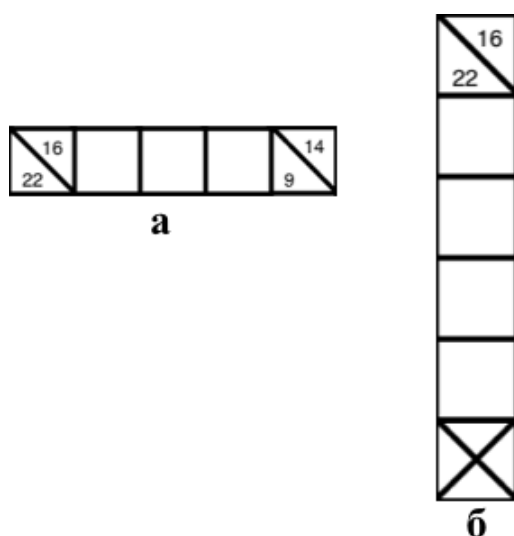


Рис. 2

Назовем такие подстроки / подстолбцы из цифровых клеток блоками. Цифры в каждом блоке не должны повторяться. Кроме того, в цифровую клетку можно записать только цифры от 1 до 9. В начале игры заполнены лишь клетки с подсказками, а все цифровые клетки пусты. Головоломка считается решенной, когда все цифровые поля заполнены таким образом, что все указанные в клетках с подсказками числа являются суммами значений соответствующих блоков.

На рис. 3 представлен решенный фрагмент поля. В горизонтальных и вертикальных блоках значения в цифровых клетках не повторяются, а сумма значений в блоках равна числам в соответствующих клетках с подсказкой.

	25	20	8	14			10	26	17	27	8	
20	2	9	4	5		26	1	6	8	9	2	
					24							
22	4	1	3	9	5	22	4	1	9	3	5	
						8						
14	8	5	1	18	8	3	5	2	3	2	1	
				27					17			
7	5	2	8	1	2	5	24	9	8	7		
			3				25				7	
28	6	3	2	8	9	28	9	8	2	5	4	
						4						
		6	1	5	11	3	8	11	7	1	3	
	20	33			26			10				
7	4	3	28	9	8	1	4	6	15	14	21	
			14									
18	1	8	2	4	3	15	3	4	1	2	5	
						8						
17	8	6	3	9	2	6	1	12	5	3	4	
				10				7				
27	5	9	1	6	4	2	15	2	3	1	9	
30	2	7	8	4	9		22	5	6	8	3	

Рис. 3

Задачей проекта была разработка игры «Какуро» для операционной системы IOS. На момент написания работы в AppStore было найдено несколько решений этой задачи:

Название	Производитель	Выбор уровня сложности	Сохранение нескольких игр	Интерактивное создание поля	Распространение
Real Kakuro™	Deucher	+	+	-	бесплатно
Primo Kakuro	JAKRO SOFT LLC	+	+	-	платно
Kakuro CS	Dogfish Software Corporation	+	-	-	платно
Kakuro +	Ralf Zimmer Zaubersee.de	-	-	-	бесплатно

Основными отличиями разрабатываемого решения стали:

- Возможность выбора одного из четырех уровней сложности. Сложность игры зависит от количества цифровых клеток на поле: чем их больше, тем сложнее игра;
- Возможность выбора размера игрового поля (10x10, 11x11, 12x12, 13x13, 14x14, 15x15);
- Сохранение и загрузка состояния игр;
- Возможность интерактивного создания игрового поля одного из перечисленных выше размеров;
- Режим обучения.

Генерация игрового поля

Поле с клетками формируется по следующим правилам: клетки в первом столбце обязательно должны быть либо пустыми, либо с подсказками. Кроме того, блок не может состоять менее чем из двух клеток и более чем из девяти. Пример игрового поля показан на рис. 4. Слева от блока (рис. 2а) и сверху блока (рис. 2б) обязательно должна быть клетка с подсказкой. В остальных случаях тип клетки генерируется случайным образом.



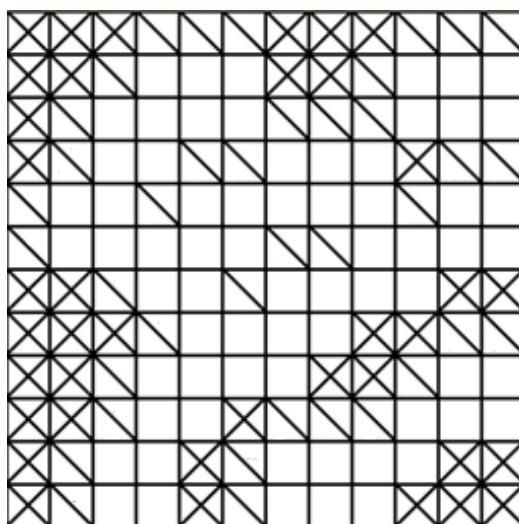


Рис. 4

Как было сказано ранее, сложность игры зависит от количества цифровых клеток на поле. В программе реализовано четыре уровня сложности: легкий, средний, сложный и очень сложный. Уровень сложности зависит от количества цифровых клеток – чем больше цифровых клеток, тем выше уровень сложности игры. Для указанных уровней сложности цифровые клетки генерируются с вероятностями: $1/8$, $3/8$, $5/8$, $7/8$ соответственно. Остальные генерируемые клетки поля будут пустыми. Все пустые клетки, которые находятся над цифровой клеткой или слева от цифровой клетки становятся клетками с подсказками.

Обычно предполагается, что поле Какуро имеет хотя бы одно верное решение. Для того чтобы создать поле, которое обязательно имело бы хотя бы одно решение, необходимо игровое поле со случайно сгенерированными типами клеток заполнить цифрами, учитывая правила игры (цифры не должны повторяться в одном блоке), т.е. решить головоломку, а уже затем поместить информацию о суммах в клетках с подсказками.

Ryan P. Davies доказал что задача заполнения игрового поля по правилам головоломки Какуро является NP-полной [2]. Класс NP – класс задач, которые можно решить на полиномиальное время на недетерминированной машине



Тьюринга. Одним из подходов к решению таких задач является метод перебора с возвратом, который использован при разработке.

Использована следующая организация данных.

Определим игровое поле прямоугольной матрицей **Game** размером ($n \times m$), где n – количество строк, m – количество столбцов. Элементы матрицы – объекты, описывающие клетки.

Введем понятие состояния цифровой клетки – это вектор, определяющий, какие цифры могут / не могут быть установлены в данный момент в клетке. Для каждого объекта цифровой клетки в матрице игрового поля зададим два вектора:

$$\text{innerNumbers}_{i,j} = [k_1 \quad \dots \quad k_9],$$

$$\text{outerNumbers}_{i,j} = [l_1 \quad \dots \quad l_9],$$

где i – номер строки матрицы **Game**,

j – номер столбца матрицы **Game**,

k_m – внутреннее состояние цифры m в клетке,

l_m – внешнее состояние цифры m в клетке,

$m = 1, 2, \dots, 9$.

Будем считать, что **outerNumbers** хранит внешнее состояние клетки, а **innerNumbers** хранит внутренне состояние клетки. Индексы элементов векторов соответствуют цифрам; если m -й элемент вектора больше 0, то цифра, равная индексу m , не может находиться в данной клетке. Например, если значение $k_2 = 0$, цифра 2 может быть вписана в клетку в текущий момент; если значение $k_2 > 0$, цифра 2 не может быть вписана в клетку в текущий момент.

Внешнее состояние клетки формируется при заполнении соседних клеток в цифровом блоке текущей строки/столбца. Внутреннее состояние клетки формируется при переборе оставшихся допустимых цифр в данной клетке. Таким образом, игровое поле представлено матрицей, каждый элемент которой является объектом класса **Brick**. Объект содержит следующие данные:

- Тип клетки;

- Цифровое значение клетки (изначально ноль);
- Горизонтальную / вертикальную суммы (изначально нули);
- Два вектора размером 9, обозначающие внешнее и внутреннее состояние клетки.

Для заполнения игрового поля цифрами необходимо выполнить следующие шаги:

1. Создать матрицу **Numbers** размером (n x m).
2. Для каждой цифровой клетки в Game записать в Numbers объекты, хранящие количество цифровых клеток, находящихся с текущей в одном цифровом блоке по горизонтали и вертикали (назовем это значение рангом клетки), и индекс текущей клетки, и отсортировать Numbers (список по строкам) по убыванию рангов.
3. Далее последовательно перебираются по строкам элементы **Numbers**. Обработываются элементы **Game** с индексами, хранящимися в объектах отсортированной матрицы **Numbers**.
 - 3.1. Если значение текущей клетки равно нулю, значит, она еще не была заполнена. В этом случае внешнее состояние клетки копируется во внутреннее.
 - 3.2. После этого проверяется вектор внутреннего состояния на наличие возможных цифр для записи в текущей клетке. Если такие цифры есть, то одна из этих цифр выбирается случайным образом. Во внутреннем состоянии клетки отмечается, что эта цифра больше не может быть выбрана (Элемент с индексом цифры увеличивается на 1). Для внешнего состояния всех цифровых клеток в блоке текущей строки и столбца отмечается, что выбранная цифра не может быть в них использована (Элемент с индексом цифры увеличивается на 1).
 - 3.3. Если все элементы внутреннего состояния стали больше нуля, т.е. доступных цифр для установки в текущей клетке нет, значит, расстановка цифр в предыдущих заполненных клетках неверна, и для текущей клетки



устанавливается значение ноль. В этом случае возвращаемся к предыдущей заполненной клетке, а цифра, находящаяся в ней, делается допустимой для установки во всех внешних состояниях цифровых клеток в блоках, в которые входит эта клетка (элемент векторов внешних состояний с индексом цифры уменьшается на 1). Во внутреннем состоянии клетки отмечается, что эта цифра использована быть не может.

Далее снова проверяется вектор внутреннего состояния и шаги алгоритма 3.1-3.3 выполняются до заполнения игрового поля. Заполнение заканчивается, когда все элементы **Numbers** рассмотрены



Поиск слагаемых для цифровых блоков

В начале игры единственная информация, которая есть у пользователя – информация о суммах в клетках с подсказками (рис. 5). Кроме этого были реализованы подсказки о возможных слагаемых для выбранной суммы при их определенном числе (рис. 6).

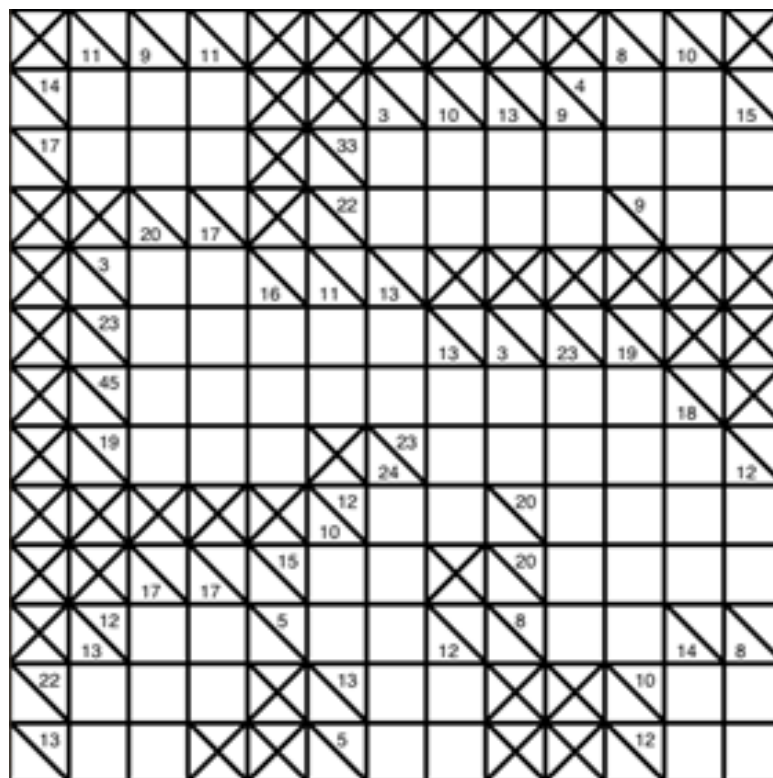


Рис. 5

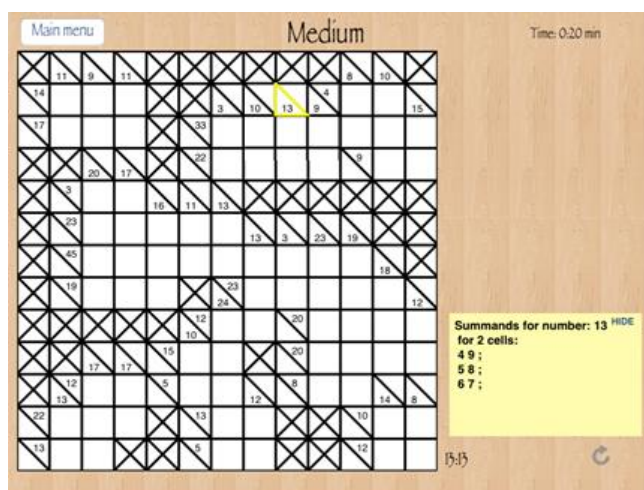


Рис. 6

Напомним, что по правилам Какуро цифры в одном блоке не должны повторяться. Для того чтобы найти все неповторяющиеся слагаемые числа, меньшие десяти, используем алгоритмом перебора с возвратом.

Обозначим искомое число C . В начале сумма S равна 1. На каждой итерации к сумме добавляется цифра, которая больше предыдущей ($1+2+\dots$). Если на некоторой итерации $C - S < 10$ & $C - S > n$, где n – последняя используемая цифра, то $C-S$ добавляется в текущий список слагаемых, а сам список запоминается. Если $C - S < 10$ & $C - S < n$, то итерация заканчивается, а вместо n проверяется $n+1$. Таким образом, каждый раз, когда проверены все возможные цифры в итерации, удаляется последняя проверенная цифра i , и проверяется $i+1$ до тех пор, пока $i+1$ не будет равным девяти включительно. Обязательным условием является то, чтобы каждое новое слагаемое было больше предыдущего – это обеспечивает их уникальность. На экран выводятся только те наборы слагаемых, в которых количество слагаемых совпадает с количеством цифровых клеток в блоке.



Сохранение и загрузка состояния игр

Решение логических головоломок может занимать довольно длительное время, и Какуро не является исключением. Кроме того, после того как один пользователь отложил решение головоломки, другой пользователь может пожелать сыграть на новом игровом поле. Для этого в приложении реализована функция сохранения и загрузки состояний до трех игр одновременно (рис. 7).

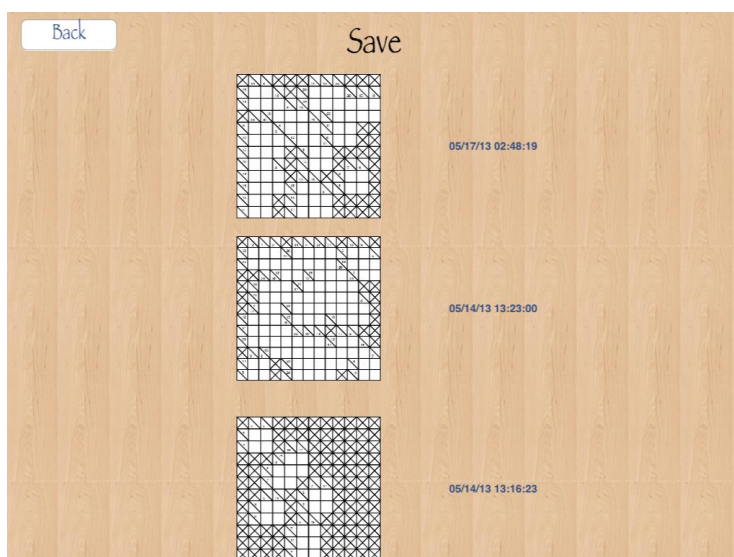


Рис. 7

Данные для каждого сохранения хранятся в трех разных файлах, которые содержат

- Скриншот игрового поля в момент сохранения;
- Закодированный массив игрового поля;
- Словарь со значением игрового таймера, датой начала игры и уровнем сложности.

В случае перезаписи ячейки сохранения у пользователя запрашивается подтверждение действия в качестве защиты от случайной потери данных сохраненных игр (рис. 8).



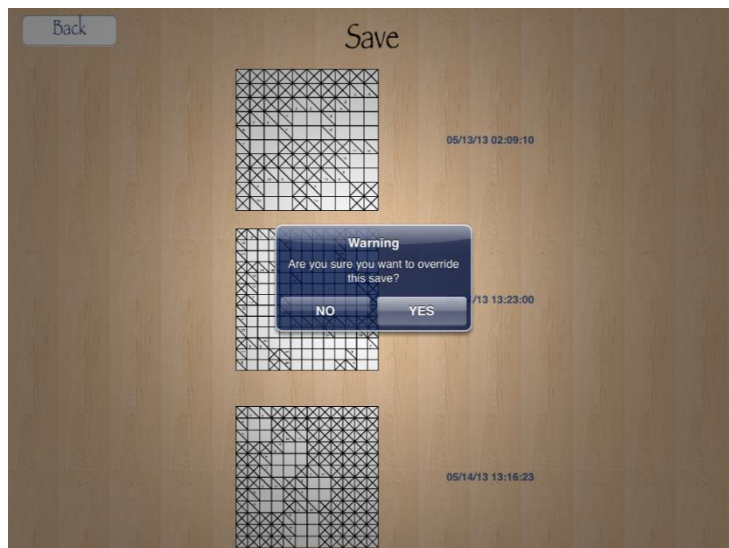


Рис. 8



Режим обучения

Для пользователей, незнакомых с правилами игры, реализован специальный режим обучения (рис. 9).

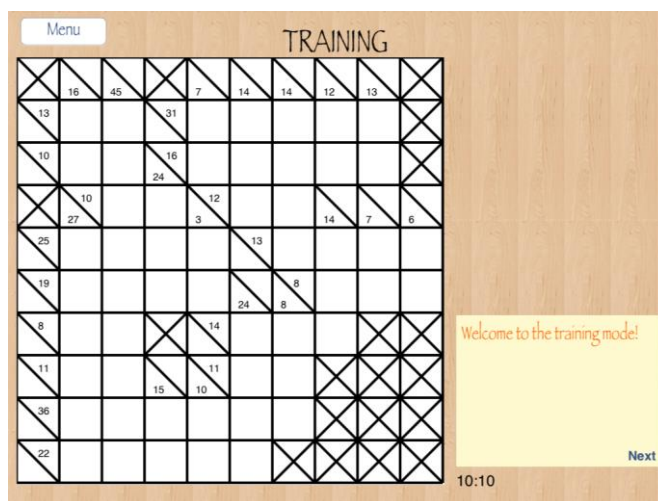


Рис. 9

Режим состоит из 10 простых шагов, в течение которых программа интерактивно демонстрирует пользователю правила игры (рис. 10).

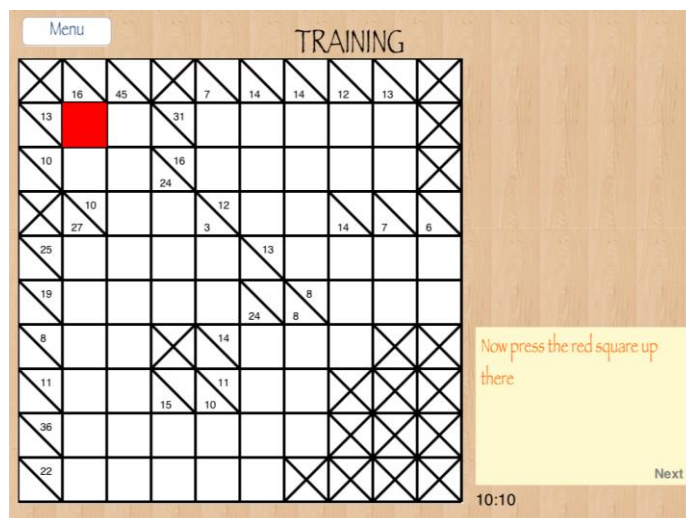


Рис. 10



Создание пользовательского игрового поля

В режиме создания пользовательского игрового поля пользователь может выбрать один из шести предложенных размеров (10x10, 11x11, 12x12, 13x13, 14x14, 15x15), после чего открывается окно создания игрового поля (рис. 11). Ограничения на левый столбец и верхнюю строку (они должны обязательно быть заполнены не цифровыми клетками) выполняется автоматически, пользователю остается только по желанию, но в соответствии с ограничениями Какуро, расставить пустые клетки.

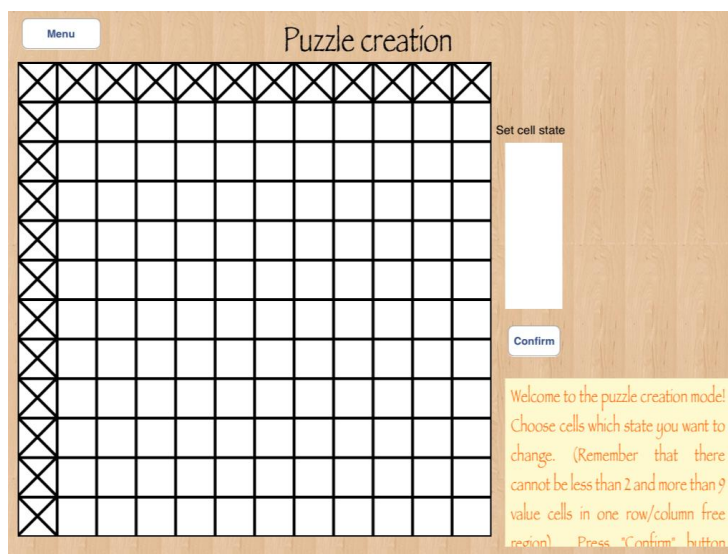


Рис. 11

Как было сказано ранее, для того чтобы расставить суммы, и головоломка имела хотя бы одно решение, необходимо сначала заполнить поле цифрами. Расставлять суммы на большом поле – довольно сложная задача, так как легко можно ошибиться. На рис. 12 показан пример небольшого поля, которое не имеет решения.

	10	10
5		
5		

Рис. 12

Таким образом, пользователь может только указать желаемое местоположение пустых клеток. Как только пользователь решает, что создание поля закончено, ему необходимо нажать на кнопку Confirm. Если поле удовлетворяет условиям Какуро (в одной строке / столбце не может быть подряд меньше двух и больше девяти цифровых клеток), то пользователю будет предложено выбрать один из пунктов меню (рис. 13):

- сыграть на созданном поле (Play this puzzle now);
- показать возможные значения цифровых клеток этого поля (Solve this puzzle);
- сохранить созданное игровое поле (Save this puzzle);
- вернуться в главное меню игры (Return to main menu).

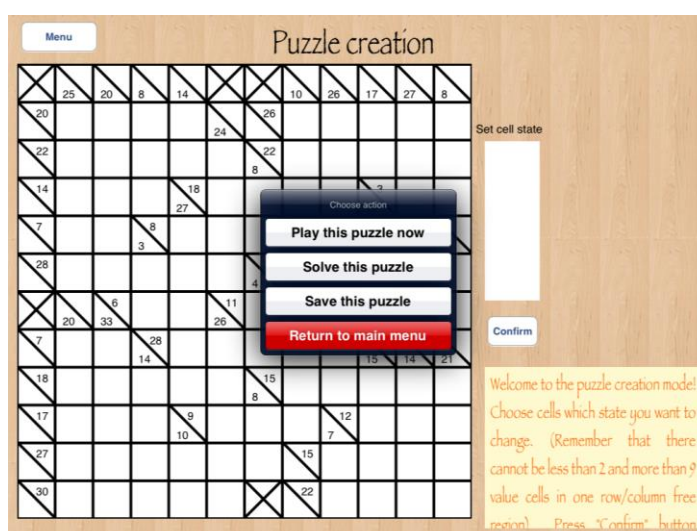


Рис. 13



Заключение

В работе были решены следующие задачи:

- Разработан алгоритм генерации клеток игрового поля в соответствии с правилами Какуро;
- Разработан алгоритм генерации числовой составляющей игрового поля в соответствии с правилами Какуро;
- Разработан интерактивный пользовательский интерфейс;
- Разработана игра Какуро, отличающаяся от существующих на момент разработки аналогов:
 - Сохранение состояний до трех игр одновременно;
 - Режим интерактивного создания уровней;
 - Режим обучения.

В приложение реализованы четыре уровня сложности игры, возможность сохранения и загрузки состояния игр, режим интерактивного создания пользователем игрового поля, режим обучения.

В дальнейшем планируется добавить в программу синхронизацию с IOS Game Center.



Список использованных источников

1. В.Рубанцев. Как решать головоломные задачи на компьютере // RVGames. — 2013.
2. Davies, R.P. (2009) An investigation into the solution to, and evaluation of, Kakuro puzzles. Unpublished M Phil thesis. University of Glamorgan. Retrieved from <http://hdl.handle.net/10265/453>
3. R.P. Davies, P.A. Roach. The Use of Problem Domain Information in the Automated Solution of Kakuro Puzzles // International Journal of Computer Science. — 2010. — стр. 118-127.
4. IOS Human Interface Guidelines: Designing for iOS 6: [Электронный ресурс]. Режим доступа: <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html> (Дата обращения 5.03.2013)

