

Разработка клеточного автомата – Игра «Жизнь» на C#

Ульянов Егор Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается и описывается разработка клеточного автомата – игра «Жизнь». Игра будет разрабатываться на языке программирования C#. Практическим результатом является разработанный клеточный автомат.

Ключевые слова: Игра «Жизнь», C#, Клеточный автомат

Development of a cellular automaton - Game of Life in C

Ulianov Egor Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses and describes the development of a cellular automaton - the game "Life". The game will be developed in the C# programming language. The practical result is the developed cellular automaton.

Keywords: Game of Life, C#, cellular automaton.

Клеточный автомат-это дискретная модель вычислений, изучаемая в теории автоматов. Клеточные автоматы также называются клеточными пространствами, автоматами тесселяции, однородными структурами, клеточными структурами, структурами тесселяции и итерационными массивами. Клеточные автоматы нашли применение в различных областях, включая физику, теоретическую биологию и моделирование микроструктуры.

Клеточный автомат состоит из регулярной сетки ячеек, каждая из которых находится в одном из конечного числа состояний, таких как ВКЛ и ВЫКЛ. Сетка может быть любой размерности. Для каждой ячейки определяется набор ячеек, называемый ее окрестностью, относительно указанной ячейки. Начальное состояние выбирается путем назначения состояния для каждой ячейки. Новое поколение создается согласно некоторому фиксированному правилу, определяющему новое состояние каждой клетки в терминах текущего состояния клетки и состояний клеток в ее окрестности. Обычно, правило обновления состояния ячеек одинаково для каждой ячейки и не изменяется во времени, а применяется ко всей сетке

одновременно, хотя известны исключения, такие как стохастический клеточный автомат и асинхронный клеточный автомат.

Эта концепция была первоначально открыта в 1940-х годах Станиславом Уламом и Джоном фон Нейманом. Хотя некоторые из них изучались в 1950-х и 1960-х годах, только в 1970-х годах и в "игре жизни" Конвея, двумерном клеточном автомате, интерес к этому предмету расширился за пределы академических кругов. В 1980-х годах Стивен Вольфрам занялся систематическим изучением одномерных клеточных автоматов. Вольфрам опубликовал новый вид науки в 2002 году, утверждая, что клеточные автоматы имеют применение во многих областях науки. К ним относятся компьютерные процессоры и криптография.

Основные классификации клеточных автоматов, описанные вольфрамом, пронумерованы от одного до четырех. Это, по порядку, автоматы, в которых паттерны обычно стабилизируются в однородность, автоматы, в которых паттерны развиваются в основном стабильные или колеблющиеся структуры, автоматы, в которых паттерны развиваются, казалось бы, хаотично, и автоматы, в которых паттерны становятся чрезвычайно сложными и могут длиться долгое время со стабильными локальными структурами. Этот последний класс считается вычислительно универсальным или способным моделировать машину Тьюринга. Особые типы клеточных автоматов являются обратимыми, когда только одна конфигурация приводит непосредственно к следующей, и тотальными, когда будущее значение отдельных клеток зависит только от общего значения группы соседних клеток. Клеточные автоматы могут моделировать различные реальные системы, в том числе биологические и химические.

Игра жизнь представляет собой клеточный автомат, разработанный британским математиком Джоном Хортоном Конвеем в 1970 году. Это самый известный пример клеточного автомата. Система Конвея - симуляция жизни, что означает, что ее эволюция определяется ее начальным состоянием, не требующим дальнейшего участия со стороны человеческих игроков. Обычно человек взаимодействует с игрой жизни, создавая начальную конфигурацию и наблюдая, как она развивается.

Правила игры «Жизни»

- Место действия этой игры — «вселенная» — это размеченная на клетки поверхность или плоскость — безграничная, ограниченная, или замкнутая (в пределе — бесконечная плоскость).
- Каждая клетка на этой поверхности может находиться в двух состояниях: быть «живой» (заполненной) или быть «мёртвой» (пустой). Клетка имеет восемь соседей, окружающих её.
- Распределение живых клеток в начале игры называется первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:
 - в пустой (мёртвой) клетке, рядом с которой ровно три живые клетки, зарождается жизнь;

- если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае, если соседей меньше двух или больше трёх, клетка умирает («от одиночества» или «от перенаселённости»)

Игра прекращается, если

- на поле не останется ни одной «живой» клетки
- конфигурация на очередном шаге в точности (без сдвигов и поворотов) повторит себя же на одном из более ранних шагов (складывается периодическая конфигурация)
- при очередном шаге ни одна из клеток не меняет своего состояния (складывается стабильная конфигурация; предыдущее правило, вырожденное до одного шага назад)
- Эти простые правила приводят к огромному разнообразию форм, которые могут возникнуть в игре.

Игрок не принимает прямого участия в игре, а лишь расставляет или генерирует начальную конфигурацию «живых» клеток, которые затем взаимодействуют согласно правилам уже без его участия.

Цель данной статьи создать клеточный автомат, а конкретнее игру «Жизнь».

В результате своей работы А.Ю. Субботина, Н. И. Хохлов предложили трехмерную версию игры Конвея «Жизнь» [1]. Н. Н. Додобоев, О. И. Кукарцева, Я. А. Тынченко рассмотрели вопросы появления различных языков программирования (в частности C#), определения особенностей этих языков, а также составления основных видов и классификаций языков программирования[2]. В своей работе Р.С. Антонов, А.В. Иванов реализовали и провели исследования блочных автоматов[3].

Создаем и называем проект. Подключаем необходимые библиотеки и создаем нужные нам переменные. Также рисуем свое поле см. рисунок 1.

```
Program.cs  Program.cs
ConsoleApp8  grid.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace grid
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             bool endgame = false;
14             int[,] grid = new int[,]
15             {
16                 { 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,},
17                 { 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,},
18                 { 0, 1, 1, 0, 1, 1, 1, 0, 0, 0,},
19                 { 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,},
20                 { 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,},
21                 { 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,},
22                 { 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,},
23                 { 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,},
24                 { 1, 1, 0, 0, 1, 0, 0, 1, 0, 0,},
25                 { 1, 1, 0, 0, 0, 0, 0, 0, 0, 1,},
26             };
27             gridconstructor gameoflife = new gridconstructor();
28             while (endgame == false)
29             {
30
31                 Console.WriteLine("Чтобы продолжить введите 1, для выхода 2:");
32                 Console.WriteLine();
33                 int choice = Convert.ToInt32(Console.ReadLine());
34                 if (choice == 1)
35                 {
36                     gameoflife.activegird(grid);
37                 }
38                 else
39                 {
40                     Environment.Exit(1);
41                 }
42             }
43         }
44     }
45 }
```

Рис. 1. Создание сетки, подключение библиотек инициализация переменных

Далее необходимо написать код копирования, проверки сетки, а также получения количества строк/столбцов в сетке см. рисунок 2.

```

47 class gridconstructor
48 {
49     int[,] livegrid;
50     int[,] oldgrid;
51
52
53     public void activegrid(int[,] grid)
54     {
55         livegrid = (int[,])grid.Clone();
56
57         int gridcol = livegrid.GetLength(0);
58         int gridrow = livegrid.GetLength(1);
59
60         for (int i = 0; i < gridcol; i++)
61         {
62             for (int j = 0; j < gridrow; j++)
63             {
64                 int livecellcount = cellchecker(i, j);
65
66                 if (livegrid[i, j] == 1)
67                 {
68                     if (livecellcount <= 1)
69                     {
70                         livegrid[i, j] = 0;
71                     }
72                     else if (livecellcount == 2 || livecellcount == 3)
73                     {
74                         livegrid[i, j] = 1;
75                     }
76                     else
77                     {
78                         livegrid[i, j] = 0;
79                     }
80                 }
81                 else
82                 {
83                     if (livecellcount == 3)
84                     {
85                         livegrid[i, j] = 1;
86                     }
87                     else
88                     {
89                         livegrid[i, j] = 0;
90                     }
91                 }
92             }
93         }
94     }
95 }
96
97
98
99

```

Рис. 2. Код копирования, проверки сетки, а также вложенный цикл для итерации через сетку

Вызываем метод для отображения изменений и повторения цикла.

```

oldgrid = (int[,])livegrid.Clone();
displaygrid(livegrid);
Console.WriteLine("Для продолжения введите у: ");
string checkkey = Console.ReadLine();
checkkey = checkkey.ToLower();
if (checkkey == "y")
{
    activegrid(oldgrid);
}

else
{
    return;
}
}

```

Рис. 3. Продолжение кода

Далее необходимо запрограммировать правила клеточного автомата. Для этого напишем метод проверки окружающих ячеек позиции на сетке (рис.4-5).

```

121 public int cellchecker(int x, int y)
122 {
123
124     int livecells = 0;
125
126
127
128     if (x < 9 && y < 9)
129     {
130         if (livegrid[x + 1, y + 1] == 1)
131         {
132             livecells++;
133         }
134     }
135
136
137     if (x < 9)
138     {
139         if (livegrid[x + 1, y] == 1)
140         {
141             livecells++;
142         }
143     }
144
145     if (y < 9)
146     {
147         if (livegrid[x, y + 1] == 1)
148         {
149             livecells++;
150         }
151     }
152
153
154     if (x > 0 && y > 0)
155     {
156         if (livegrid[x - 1, y - 1] == 1)
157         {
158             livecells++;
159         }
160     }
161 }

```

Рис. 4. Программирование правил автомата

```

162
163     {
164         if (livegrid[x, y - 1] == 1)
165         {
166             livecells++;
167         }
168     }
169
170     if (x > 0)
171     {
172         if (livegrid[x - 1, y] == 1)
173         {
174             livecells++;
175         }
176     }
177
178     if (x < 9 && y > 0)
179     {
180         if (livegrid[x + 1, y - 1] == 1)
181         {
182             livecells++;
183         }
184     }
185
186     if (x > 0 && y < 9)
187     {
188         if (livegrid[x - 1, y + 1] == 1)
189         {
190             livecells++;
191         }
192     }
193
194     return livecells;
195 }
196
197
198 public void displaygrid(int[,] livegrid)
199 {
200     int gridcol = livegrid.GetLength(0);
201     int gridrow = livegrid.GetLength(1);
202
203     for (int i = 0; i < gridcol; i++)
204     {
205         for (int j = 0; j < gridrow; j++)
206         {
207             Console.Write("{0,2}", livegrid[i, j]);
208         }
209         Console.WriteLine();
210     }
211 }
212
213 }

```

Рис. 5. Продолжение кода

Теперь игра «Жизнь» готова и на рисунках ниже представлена ее реализация см. рисунок 6.

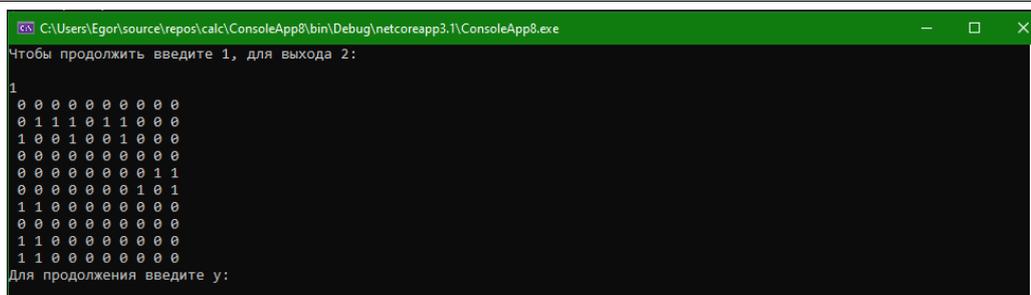


Рис. 6. Сетка игры

Для дальнейших итераций жмем клавишу у и Enter см. рисунок 8.

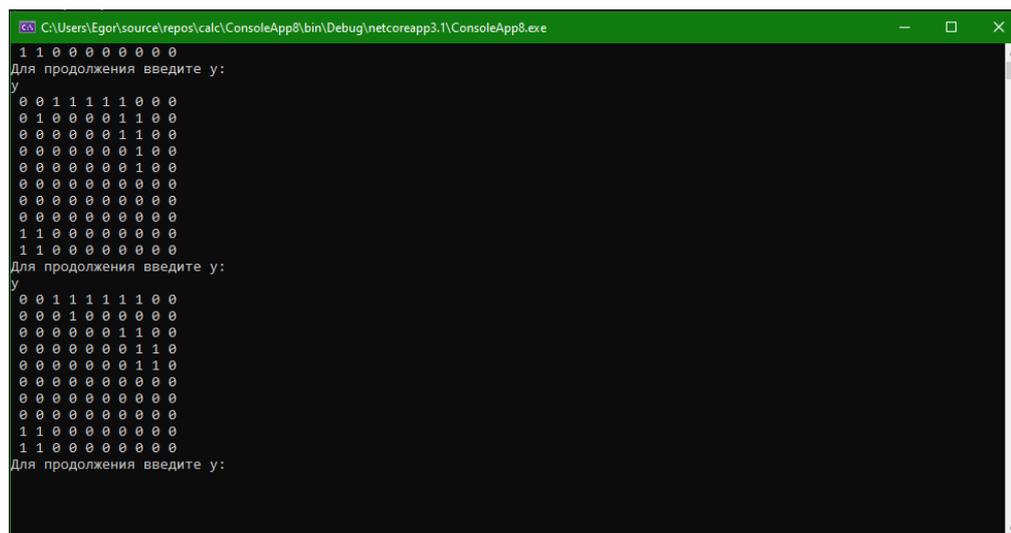


Рис. 7. Поведение клеток после двух итераций

Как видно, алгоритм работает и имеет потенциал к дальнейшему развитию.

В данной статье был создан клеточный автомат – игра «Жизнь» в среде разработки Visual Studio.

Библиографический список

1. Субботина А.Ю., Хохлов Н. И. Реализация клеточных автоматов "игра жизнь" и клеточного автомата Кохомото-ооно с применением технологии MPI // Компьютерные исследования и моделирование. 2010. №17. С. 319-322.
2. Додобоев Н. Н., Кукарцева О. И., Тынченко Я. А. Современные языки программирования // Современные технологии: актуальные вопросы, достижения и инновации. 2014. №5. С. 81-85.
3. Антонов Р.С., Иванов А.В. Моделирование блочных клеточных автоматов // Проблемы управления в социально-экономических и технических системах. 2019. №9. С. 320-334.