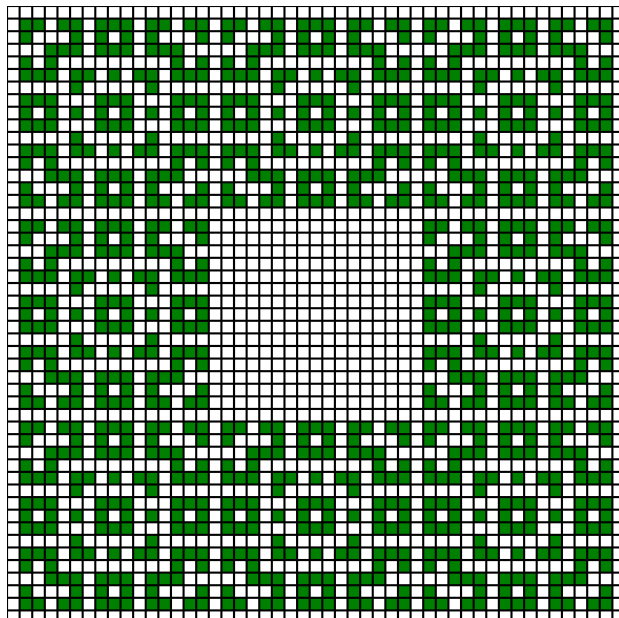


Живые Пиксели. Эволюция.



На рисунке слева показан результат развития одной активной клетки алгоритма «Живые Пиксели» за 23 шага, справа – результат развития белковых тел за 4 миллиарда лет

В работе изучается простейший из возможных алгоритмов.

Основные гипотезы и находки изучаемого алгоритма

- Живые Пиксели - это простейший возможный алгоритм, дающий интересные результаты и исследовать его необходимо, потому что наука стремится изучать любые простейшие, базовые структуры во всех областях.
- График изменения числа пикселей, выходящий из одного начального, подчиняется философским законам Гегеля. Возможно, это может служить неявным их доказательством.
Изучение свойств математических рядов, получающихся из числа пикселей, растущих из одного начального.
- Размножение фигур. Данное свойство ранее было описано американским проф. Фредкиным.
- Обратимость. Некоторые матрицы обладают свойством обратимости — любая фигура через определённое количество шагов возвращается в первоначальное состояние. Свойство может быть использовано на практике для шифрования.
- Другие интересные находки.

Живые Пиксели. Эволюция.

Кто управляет нашей жизнью? На каких законах построено развитие жизни – от простейших одноклеточных организмов до носителей разума? Возможно ли, чтобы жизнь на планете самостоятельно достигла таких сложно взаимодействующих структур, если нет внешнего управления и всё происходит либо случайно, либо на основании базовых законов физики и теории вероятностей?

Теория Дарвина говорит об эволюционном развитии жизни: за счёт инстинктов, борьбы за выживание и генетической наследственности. Проверить такую теорию на практике очень сложно, ведь эволюция идёт миллиарды лет, и человек не может смоделировать её, к примеру, на подопытных животных, ввиду недостатка времени.

Идея «Живых Пикселей» заключается в поиске элементарного алгоритма, который бы давал на выходе сложные структуры – чтобы смоделировать появление и развитие живых существ исходя из элементарных алгоритмов, которые могут образоваться случайным образом. Практическое применение системы возможно для развития Искусственного Интеллекта – по большому счёту Искусственный Интеллект только тогда станет Интеллектом, когда начнёт развиваться самостоятельно, и именно такое самостоятельное развитие моделируют «Живые Пиксели».

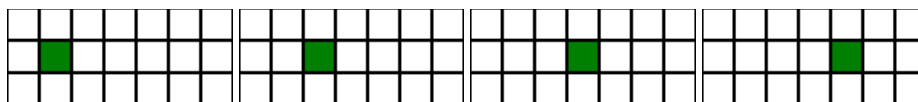
«Живые Пиксели» показывают, как из тетрадного листа в клетку и простого логического правила рождается сложный мир, имеющий интересные сходства с реальностью. Здесь Вы увидите простейший алгоритм размножения, большой взрыв вселенной, закон перехода количества в качество, сохранение наследственной информации при делении ДНК, эволюцию видов и многое другое.

Живые Пиксели — это сложнейший мир, рождённый всего из одного правила

Живые Пиксели — это так называемый «клеточный автомат» - последовательное изменение рисунка, состоящего из активных и пустых клеток, где каждый новый образ создаётся из предыдущего по определённым правилам.



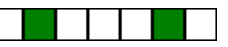

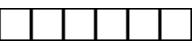
Алгоритм «Живые Пиксели» реализован в бесплатной программе «NeoNeuro Живые Пиксели – Эволюция», которую можно скачать на сайте www.neoneuro.com

Рассмотрим пример – каждая активная клетка делает активной правую и сама исчезает. На следующем наборе рисунков показано последовательное изменение клеточного автомата по такому правилу:



В данном случае мы видим «движение» активной клетки (пикселя) вправо.

Пиксель в этом примере движется по одной горизонтали, другие «строки» для алгоритма необязательны. Такой случай является одномерным и может быть представлен одной строкой. Рассмотрим правило на одномерном пространстве, когда активный пиксель меняет своё состояние и состояние своих соседей на противоположное. Такое изменение называется инвертированием.

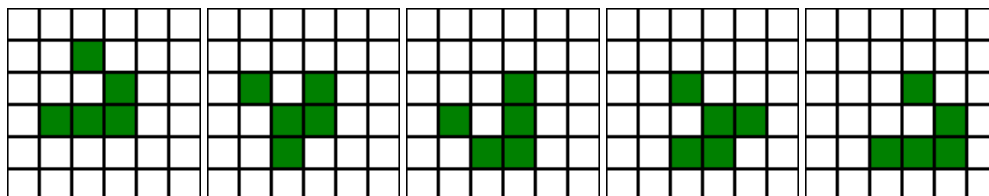
				
начальное положение	шаг 1	шаг 2	шаг 3	шаг 4

На первом шаге вместо центрального активного пикселя мы видим два новых – это его «соседи». На третьем шаге существующие пиксели исчезли, на их месте появились новые соседи, при этом начальный пиксель не вернулся, хотя и являлся соседом обоих пикселей из первого шага. Почему? Потому что первый «сосед» сделал пиксель активным, второй – обратно вернул его в пустое состояние - инвертировал. Данное правило иногда называют XOR алгоритмом и оно является одномерным случаем главного рассматриваемого в статье алгоритма – «Живые Пиксели». На третьем шаге мы видим четыре активных пикселя, на четвёртом все пиксели исчезли. Здесь рассматривается случай, где пространство ограничено справа и слева, если взять неограниченное пространство, то по такому алгоритму будет происходить бесконечное «распространение» активных клеток в обе стороны.

Игра Жизнь

«Игра Жизнь» - это наиболее известный клеточный автомат, разработанный английским математиком Джоном Конвеем в 1970м году. Она базируется на следующих правилах:

- пустая клетка, рядом с которой находятся ровно три живые клетки, оживает;
- 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».



На изображении показана наиболее известная фигура – планер (glider), фигура видоизменяется на каждом шагу, и через четыре этапа мы опять видим начальный рисунок, но смещённый на одну клетку вправо-вниз.

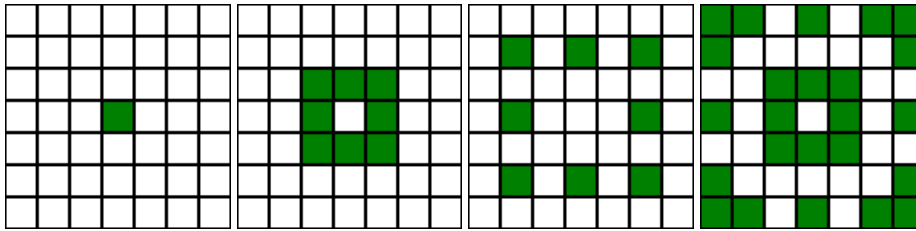
В данном алгоритме большинство начальных позиций либо быстро исчезает, либо перерастает в набор «простых» элементов, при этом можно создать конструкции, обладающие интересными свойствами – например, конструкции, «перекидывающие» друг другу «мячик» или «ружьё», «стреляющее» планерами. Множество учёных и энтузиастов создавало и продолжает создавать сложные интересные фигуры в «Игре Жизнь».

В теории клеточных автоматов правило можно записать как B3/S2, где B: born – рождённые, S: survival – оставшиеся, цифры означают количество соседей. То есть если активных соседей три, то рождается новый элемент, если соседей два – то элемент продолжает существовать, во всех прочих случаях клетка остаётся или становится пустой.

Алгоритм «Игра Жизнь» с примерами можно найти в программе «NeoNeuro Живые Пиксели».

Живые Пиксели

В Живых Пикселях правило только одно — активная клетка меняет своё состояние и состояние всех 8 соседних на противоположное. Другими словами – активная клетка инвертирует квадрат 3 x 3, внутри которого она находится.



Активный пиксель в центре меняет состояние пустых клеток рядом на активное. Сам становится пустым, потому что активных соседей у центрального пикселя в начальный момент нет. Изображения слева направо показывают последовательное изменение рисунка.

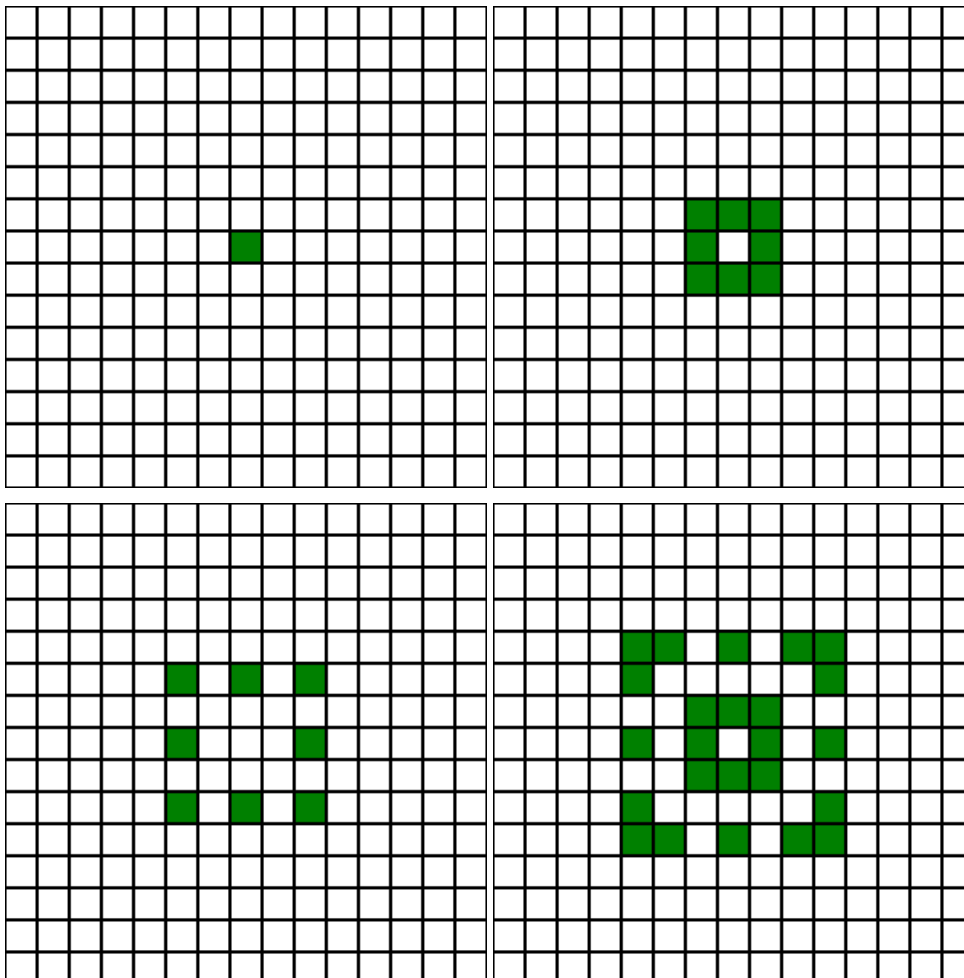
Равносильное правило — если для клетки число активных соседей нечётное, она становится активной, если чётное — то пустой. В теории клеточных автоматов правило можно записать как B1357/S1357 – B: born – рождённые, S: survival – оставшиеся, цифры означают количество соседей.

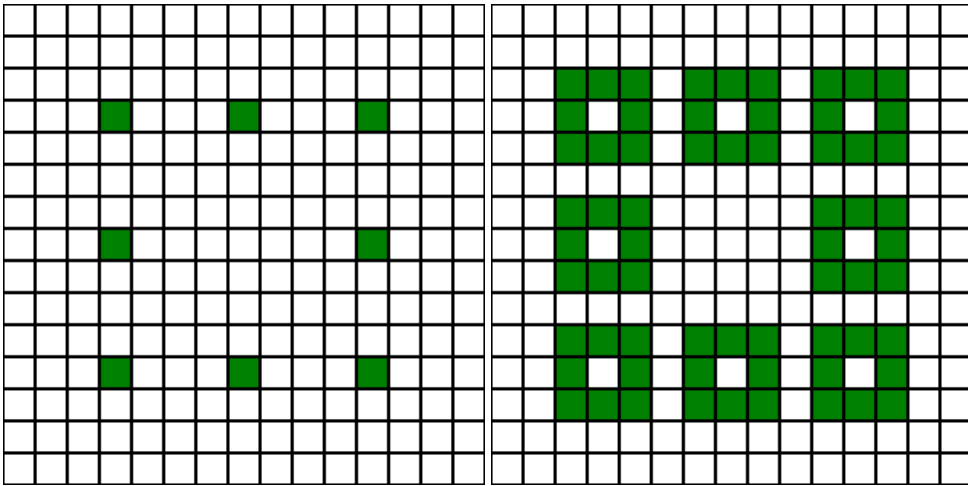
Если взять ограниченное квадратное поле размером два на два и поместить туда один активный пиксель, то получатся два циклически повторяющихся варианта:



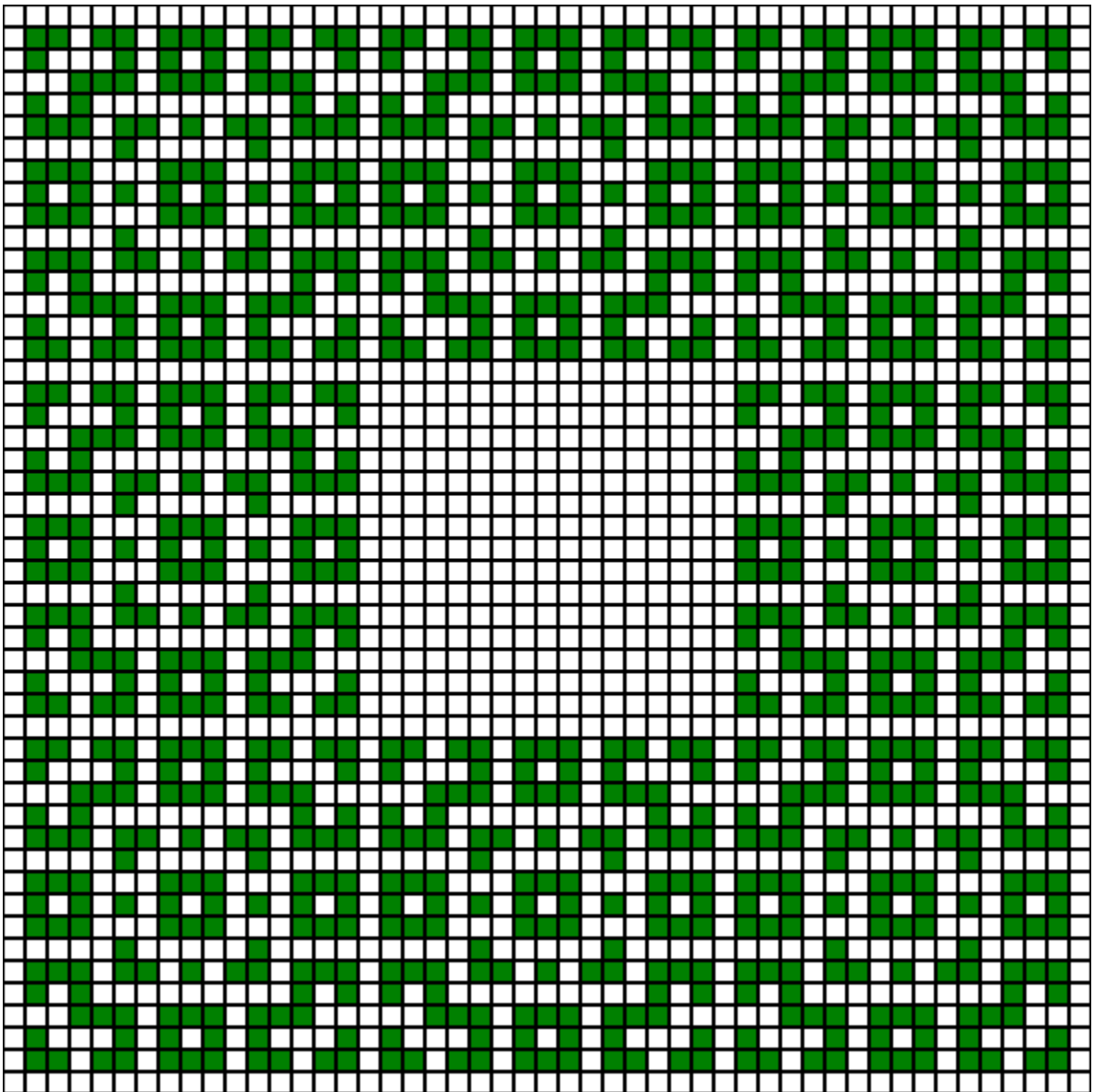
Начальный пиксель делает активными три других, затем три активных пикселя делают активным начальный, и сами становятся пустыми, так как у каждого ДВА «живых» соседа — чётное число, значит клетка будет пустой.

На следующем рисунке показано, как один активный пиксель эволюционирует в красивый арабский орнамент.





На двадцать третьем шаге видим следующую картину, появившуюся «случайно», исходя из одной активной клетки и простого алгоритма, вместе с тем причудливый узор настолько сложен и красив, что возникает мысль о том, что он создан человеком.



В Живых Пикселях рассматривается, в первую очередь, двумерная матрица и «расширенное» количество соседей — 8 штук, ход шахматного короля, в теории клеточных автоматов 8 соседних клеток называются «окрестностью Мура». При этом алгоритм актуален и показывает интересные возможности и на одномерной, трёхмерной и многомерной матрице, а также может быть использован для «укороченного» числа соседей — 4 клеток, без учёта диагональных. В Живых Пикселях поле считается ограниченным, при этом неограниченную матрицу можно считать частным случаем ограниченной со «стенками», удалёнными на бесконечное расстояние. В программе проанализировано большое количество различных начальных положений активных пикселей — так называемых паттернов, большое внимание уделено простейшему паттерну — одной живой клетке на бесконечной матрице.

В чём заключается уникальность алгоритма Живые Пиксели?

Чем он отличается от других клеточных автоматов, почему рассматривается именно случай с инвертированием активных клеток? Ответ заключается в том, что это наиболее простой алгоритм, который позволяет имитировать развитие вселенной и сложных форм жизни. Рассмотренный ранее алгоритм с смещением пикселя вправо не даёт интересных свойств. Алгоритм «Жизнь» очень интересен, и всё же его правила сложнее, чем единственное правило Живых Пикселей:

«Активная клетка меняет своё состояние и состояние соседних клеток на противоположное».

Простое правило даёт удивительные и неожиданные открытия, которые далее будут детально проанализированы:

- Один активный пиксель рождает красивый фрактал, имитирующий Большой Взрыв Вселенной.
- Изменение количества активных пикселей в паттерне «Вселенная» подчиняется философскому закону перехода количества в качество и рождает уникальный математический ряд
- Размножение фигур. Любая начальная конструкция из активных пикселей на бесконечном поле будет со временем делиться и размножаться с сохранением наследственной информации.
- Обратимость. Некоторые матрицы обладают свойством обратимости — любая фигура через определённое количество шагов возвращается в первоначальное состояние. Свойство может быть использовано на практике для шифрования.

Анализ алгоритма Живые Пиксели

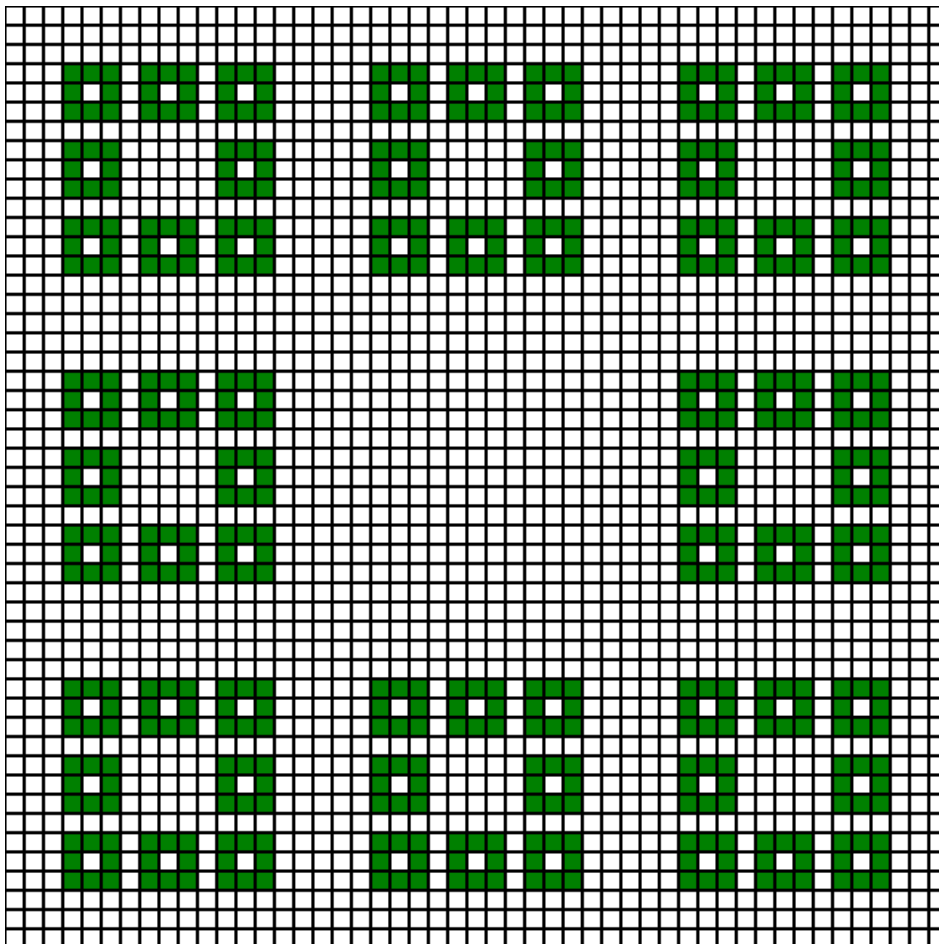
Разберём различные варианты, которые даёт алгоритм Живых Пикселей - различные сочетания активных и пустых пикселей, одномерную и двумерные ограниченные и неограниченные матрицы.

Все указанные примеры, а также множество других можно найти в программе.

Очень интересным представляется анализ развития автомата из одной активной клетки на бесконечном поле. В первом приближении этот вариант уже был рассмотрен ранее. Он называется «Вселенная» ввиду определённых сходств с развитием нашей Вселенной в результате большого взрыва.

Вселенная. Большой Взрыв.

"Живые Пиксели" - система изменения пикселей, когда каждый активный пиксель меняет состояние соседних пикселей на противоположное: активный становится пустым, а пустой - активным. Простое правило образует очень сложные конструкции. Один живой пиксель на бесконечном поле имитирует Большой Взрыв Вселенной.



На 21м ходу картинка представляет собой квадрат, состоящий из 8ми квадратов, каждый из которых также состоит из 8ми квадратов, каждый из которых состоит из 8ми пикселей. Такое подобие называется Фрактал, наша Вселенная также имеет структуру Фракталя: Галактики вращаются вокруг центра Вселенной, Звезды вращаются вокруг центров Галактик, Планеты вокруг Звёзд, Электроны вокруг Ядер атомов. Изменение числа активных пикселей на каждом шагу образует очень интересный математический ряд. В таблице последовательность чисел представлена справа налево и далее сверху вниз по рядам:

1	8	8	24	8	64	24	112	8	64	192	24	192	112	416
8	64	64	192	64	512	192	896	24	192	192	576	112	896	416
1728		8	64	64	192	64	512	192	896	64	512	512	1536	

Табличные данные удобно представить в виде диаграммы. По вертикали – количество активных пикселей, по горизонтали – шаг.



С одной стороны, число пикселей постоянно увеличивается, с другой - их число неизменно обращается 8! При работе программы такое обращение кажется волшебным: из причудливого зелёного тумана выходит новорожденный - те же 8 пикселей, но на большем расстоянии друг от друга. Такое преобразование подчиняется философскому закону "перехода количества в качество".

Закон перехода количественных изменений в качественные

Закон был выведен Гегелем и затем сформулирован Энгельсом. Вот что о нём пишет Wikipedia:

Предметом закона является переход от незначительных и скрытых, постепенных количественных изменений к изменениям коренным, открытым — качественным, где качественные изменения наступают не случайно, а закономерно, вследствие накопления незаметных и постепенных количественных изменений, не постепенно, а быстро, внезапно, в виде скачкообразного перехода от одного состояния к другому состоянию, через ломку линейного закона изменения и перехода к нелинейным законам и формам изменения.

Момент перехода от накопления количественных изменений к качественным называется скачок. Например, это точка кипения воды — при 100 градусах количественные изменения температуры воды превращают её в пар — в новое качество. Это всеобщий закон, которому подчиняются процессы изменения в самых разных областях, таких как физика, химия, техника, социальные и политические явления, экономика и других.

Паттерн Вселенная хорошо вписывается в закон перехода количества в качество — на графике отлично видно, как на 7м шаге количество активных пикселей доходит до критической точки — 112 штук, после чего идёт качественный скачок, и на 8м шаге мы видим 8 пикселей. На 14м шаге наблюдаем 416 пикселей, и опять скачок — на следующем шаге их снова 8.

Ещё раз приведём диаграмму изменения числа живых пикселей, для большего числа шагов:



График на диаграмме можно назвать функцией изменения числа живых пикселей, с другой стороны – это график биения нашей жизни, он показывает зависимости, которым подчиняется всё живое и неживое в природе.

Обратите внимание на поведение графика с 16го по 23й шаг. Сравните со значениями на шагах с 32го по 39й. Поведение графика в данных промежутках абсолютно идентичное – все значения совпадают! Теперь это напоминает третий закон диалектики Гегеля – закон отрицания – который утверждает, что жизнь развивается по спирали – всё в мире возвращается «на круги своя», при этом с небольшими качественными изменениями.



Спиралевидная галактика

Рассмотрим ряд, когда происходят качественные скачки. В таблице представлены номера шагов, которым соответствуют 8 живых пикселей.

1	2	4	8	16	32	64	128
---	---	---	---	----	----	----	-----

Мы видим главный «программистский» ряд, состоящий из круглых чисел в двоичной системе: 1, 10, 100 и т.д. Именно в эти моменты Живые Пиксели показывают качественный скачок – и именно в этих цифрах заключаются основные скачки в развитии компьютеров. Например, переход от 8-битных систем к 16-битным, затем к современным 32х и 64х битным платформам вызывал и вызывает качественные изменения в архитектуре как аппаратного, так и программного обеспечения.

Рассмотрим ряд изменения числа пикселей в паттерне Вселенная, представленный в двоичном виде:

1	1000	1000	11000	1000	1000000	11000
1110000	1000	1000000	11000000	11000	11000000	1110000
110100000	1000	1000000	1000000	11000000	1000000	1000000000

Большинство чисел имеют вначале единицы, затем нули. Ещё один ряд, составленный из количества единиц в вышеуказанном ряде:

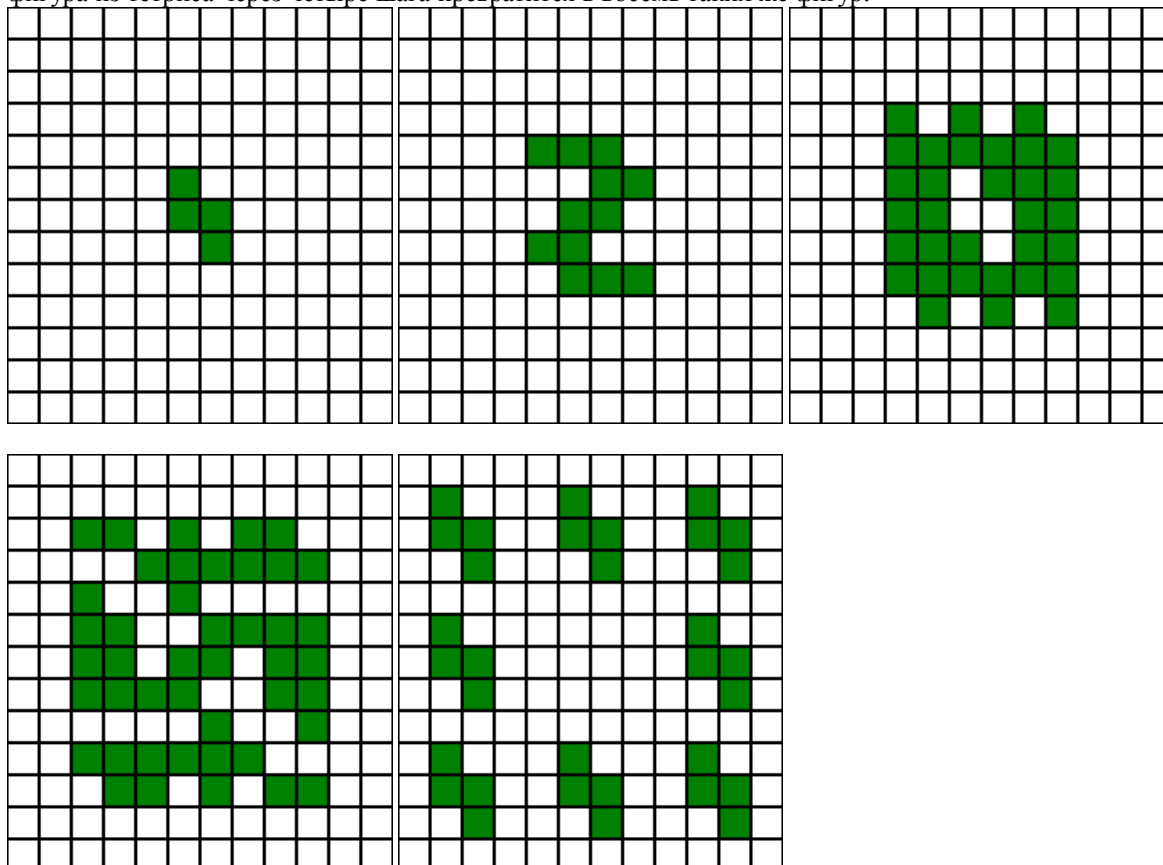
1	1	1	2	1	1	2	3	1	1	2	2	3	3	1	1	1	2	1	1	2	3	2	2	2	2	3	3	3	4	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Если применить его к клавишам фортепьяно, то получается гармоничная музыка...

Размножение.

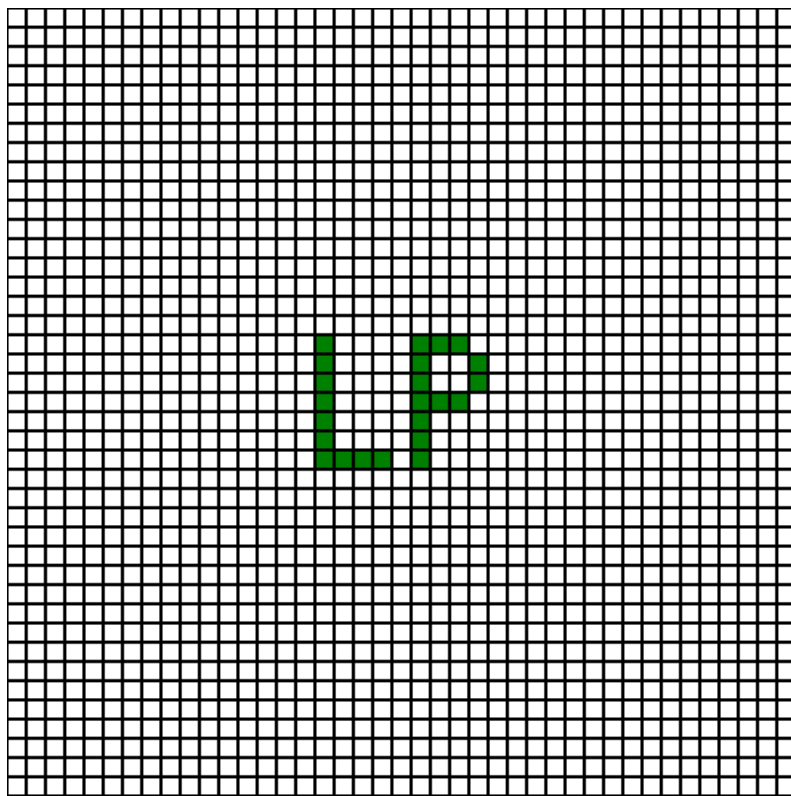
На бесконечном поле любое начальное положение активных пикселей – паттерн – через конечное число ходов повторится, причём сразу восемь раз.

Размножение является очень интересной особенностью Живых Пикселей. Давайте посмотрим, как простая фигура из тетриса через четыре шага превратится в восемь таких же фигур:

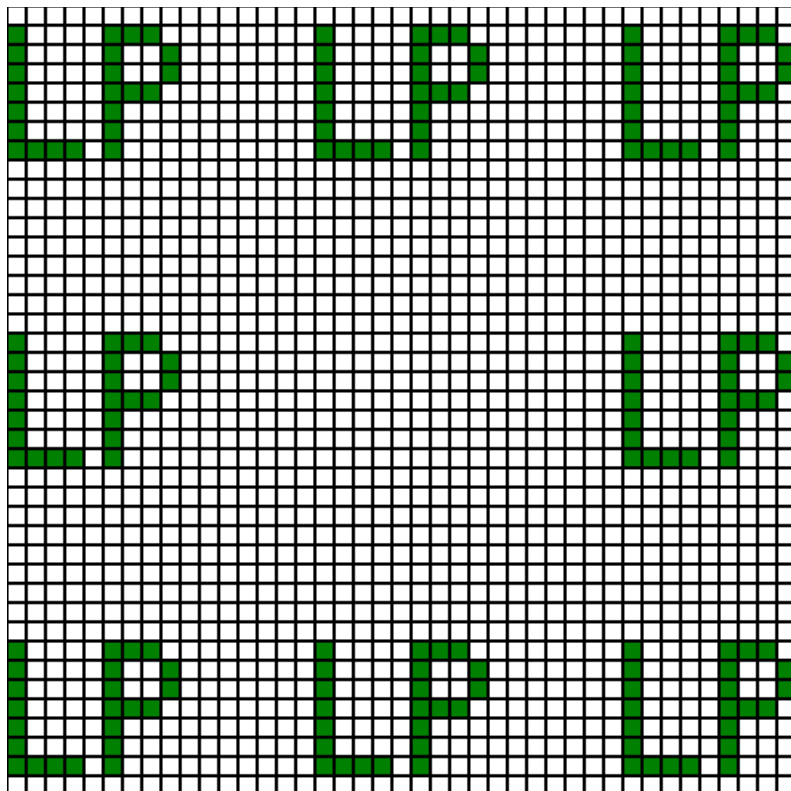


На первом шаге появляется буква Z, на втором видим «инвертированную» начальную фигуру на фоне активных пикселей, и сложный узор третьего шага сменяется «новорожденными» фигурами – такими же, как в начальной позиции.

В следующем примере знак LP «размножается» за 16 ходов:



Начальное положение

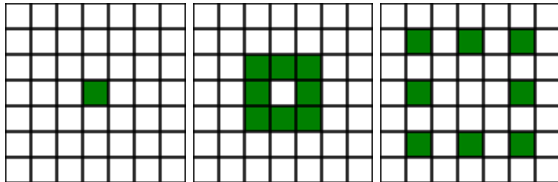


Через 16 ходов

Данный эффект был описан американским профессором Эдвардом Фредкиным.

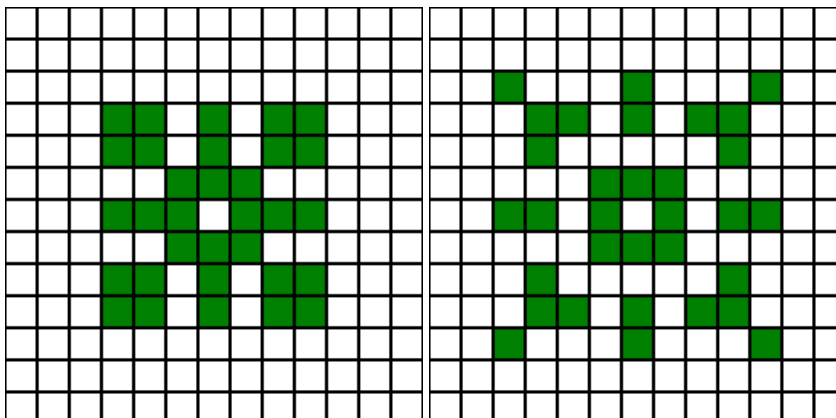
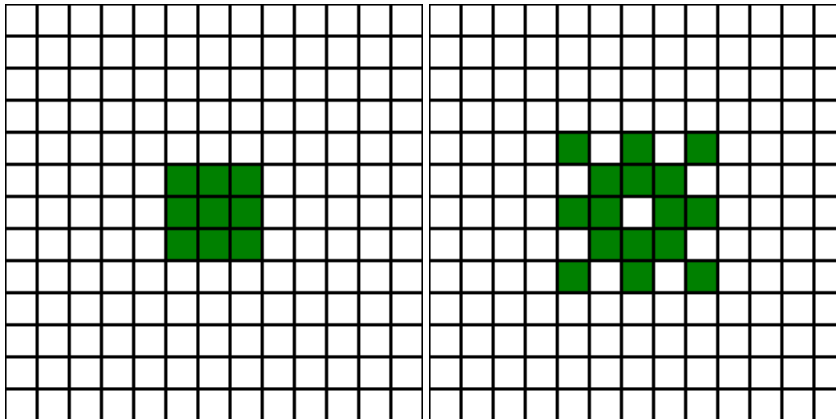
Период размножения

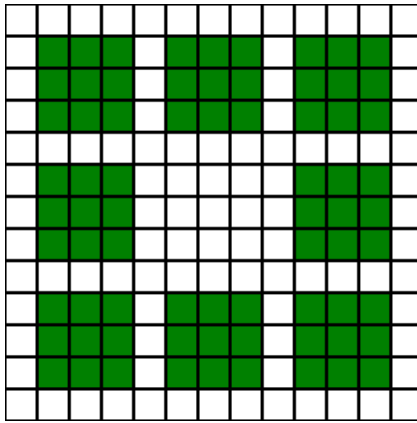
Сколько шагов требуется, чтобы начальный паттерн «размножился»? Если взять паттерн из одного пикселя, то уже на первом ходу будет восемь новых – но между ними не будет пространства, которое появляется лишь на втором шаге:



Таким образом, для простейшего паттерна период самовоспроизведения равняется двум.

В программе легко увидеть, что для квадратного паттерна 2x2 период равен четырём. Рассмотрим паттерн 3x3:





Здесь мы наблюдаем интересные рисунки, похожие то на мишень, то на людей за компьютерами в офисе, если смотреть сверху, как на предпоследнем изображении. Что касается периода размножения, то он также равен четырём, как и для паттерна 2x2. Для паттерна 4x4 он равен 8ми и так далее.

Чтобы рассчитать период размножения, обозначим длину максимальной стороны паттерна по вертикали или по горизонтали как Length.

Чтобы гарантировать пространство между отдельными новорожденными паттернами к нему необходимо добавить один. Получившееся значение увеличиваем до ближайшего числа из ряда 2, 4, 8, 16.., т.е. 2^x .

Знак LP в вышеуказанном примере имеет ширину 9 пикселей и высоту 7 пикселей. Значит, Length равен 9. Увеличиваем на один:

$$9 + 1 = 10.$$

Для 10 период равен 16 – это ближайшее к 10 значение из ряда 2, 4, 8, 16, 32... в сторону увеличения.

Рассмотрим простейшие случаи Живых Пикселей в ограниченном пространстве.

1 Ряд

Матрицы, состоящие из одного ряда также можно назвать одномерными.

Одна клетка

Пустой Пиксель - ничего не происходит. 

Активный Пиксель - становится пустым на первом ходу. 

Две клетки

В первом случае наблюдаем маятник - последовательное движение живого Пикселя вправо-влево.



Во втором случае два Живых Пикселя составляют простейшую статическую фигуру, то есть фигуру с живыми пикселями, которая сохраняет свое предыдущее состояние на каждом шагу.

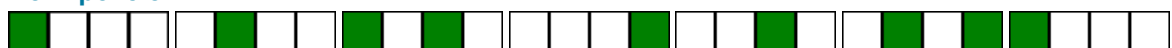


Три клетки

Все фигуры исчезают. В дальнейшем мы рассмотрим «нежизнеспособные» случаи, ведущие к исчезновению любых паттернов более детально.



Четыре клетки

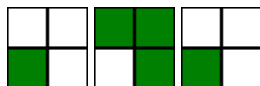


обладают уникальными свойствами, например, возможностью полного обращения рисунка в начальное состояние через определённое количество шагов.

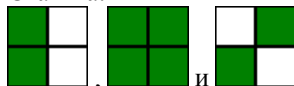
2x2

Квадрат 2x2 содержит два "мигающих" варианта, переходящих один в другой, и три статических.

Мигание:

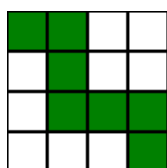
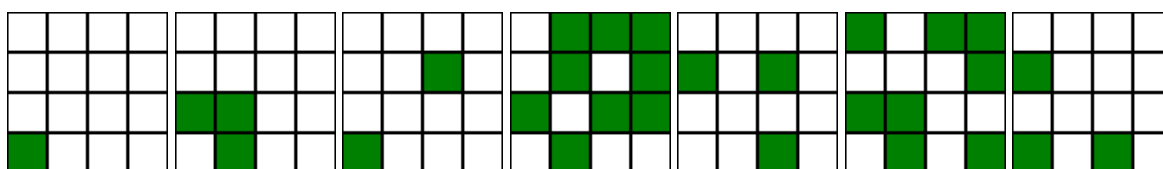


Статика:



4x4

В данной матрице встречаются фигуры, обладающие обратимостью. При этом большинство фигур необратимы, через несколько шагов они превращаются в обратимые фигуры с периодом 6 - у некоторых фигур период обращения равен 3, ввиду симметричности.

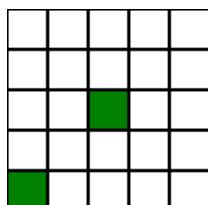
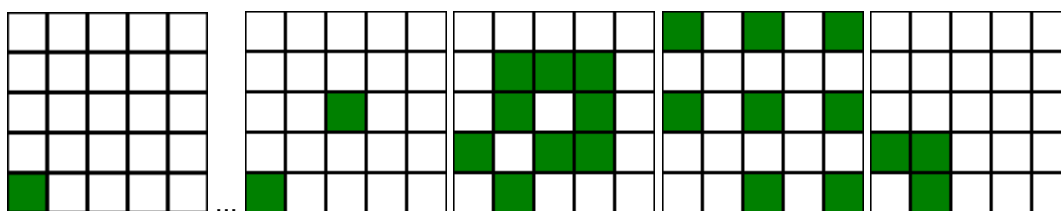


3x3

Все варианты исчезают.

5x5

Пиксель в углу превращается в конструкцию с периодом обращения 4



6x6

Полная обратимость за 14 ходов

8x8

Полная обратимость за 14 ходов

Обратимость чётных квадратов

Обратимость некоторых квадратов является интересным математическим свойством, проанализируем его в таблице.

Таблица обратимости чётных квадратов:

Размер	2x2	4x4	6x6	8x8	10x10	12x12	14x14	16x16	18x18	20x20
Период	2/0*	6/3/нет	14	14	62	126	30/нет	30/нет	1022	126

Размер	22x22	24x24	26x26	28x28	30x30	32x32	34x34	36x36	38x38	40x40
Период	4094	2046/нет	1022	32766	62	62	8190/Нет	524286	8190	2046

*дробь означает, что для некоторых паттернов – начальных рисунков – обратимости нет, либо существуют статические неизменные фигуры, с обратимостью 0, либо есть частные случаи обратимости в меньшее количество шагов, что случается для некоторых симметричных фигур.

Как видно, размеры матриц и периоды обращения соответствуют количеству клеток в одномерных матрицах и периоду обращения в них.

Период обращения представляет собой ещё один математический ряд:

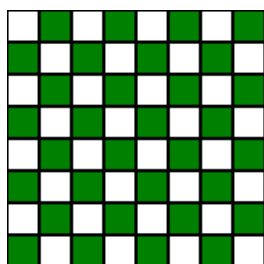
2	6	14	14	62	126	30	30	1022	126	4096
2046	1022	32766	62	62	8190	2046				

Все числа ряда являются числами, входящими в множество $2^x - 2$, где x – целое число.

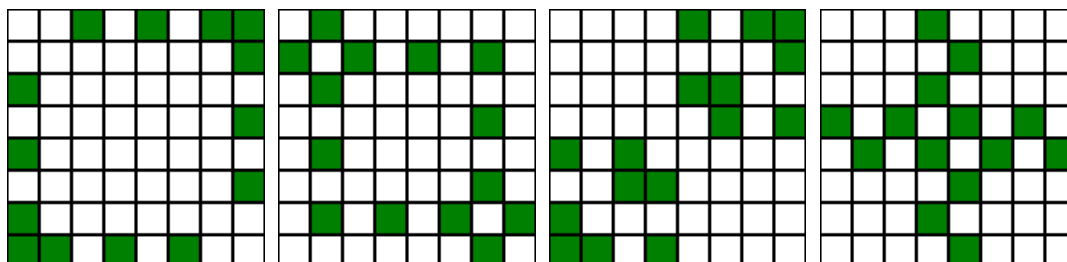
Чётные квадраты могут быть использованы на практике для шифрования. Найти "обратный" алгоритм, чтобы выяснить предыдущее состояние паттерна «Живых Пикселей» очень сложно, не исключено что практически невыгодно. При этом "прямой прогон" любого изображения в обратимом чётном квадрате даст начальное изображение через большое, заранее известное число итераций. Не буду вдаваться в детали теории шифрования и оставляю вопрос конкретных методик использования для этих целей «Живых Пикселей» открытым.

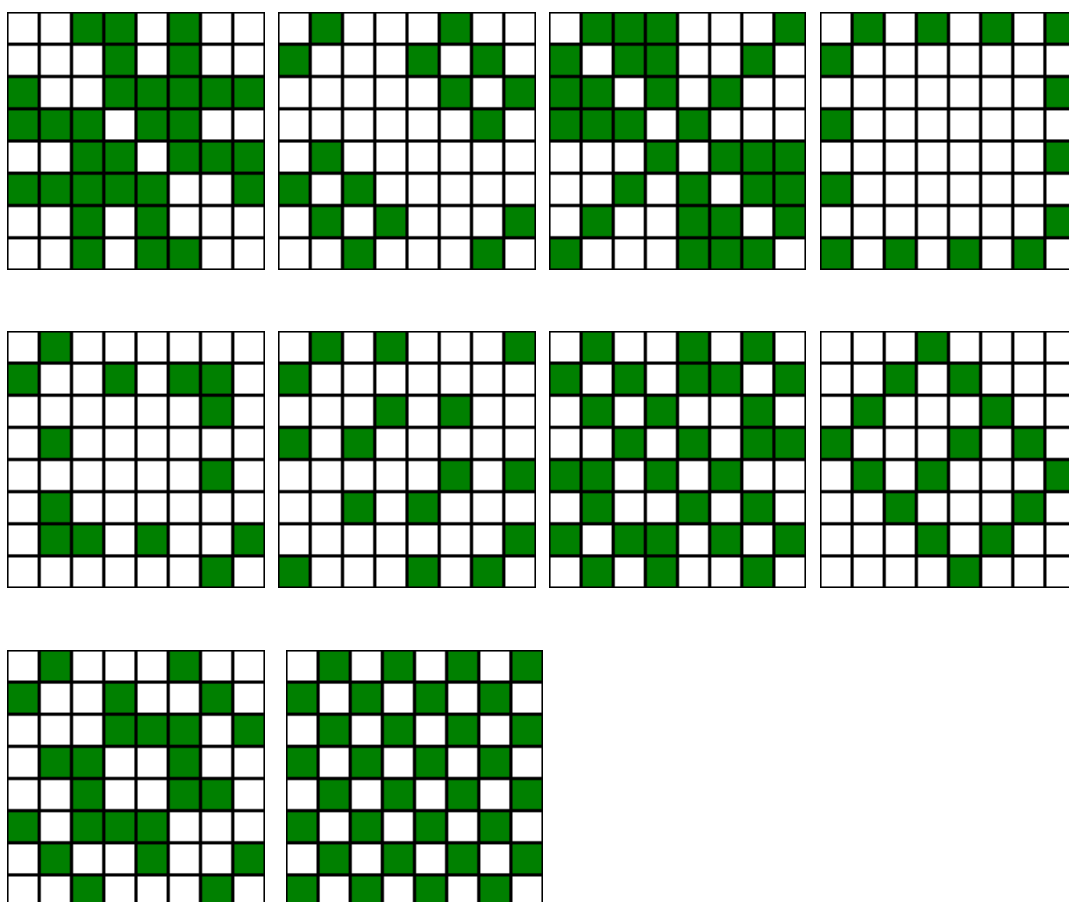
Шахматы

Рассмотрим практический случай квадрата 8x8 – шахматную доску.



Появляющиеся фигуры столь же изящны и неожиданны, как комбинации Таля в матче с Ботвинником.



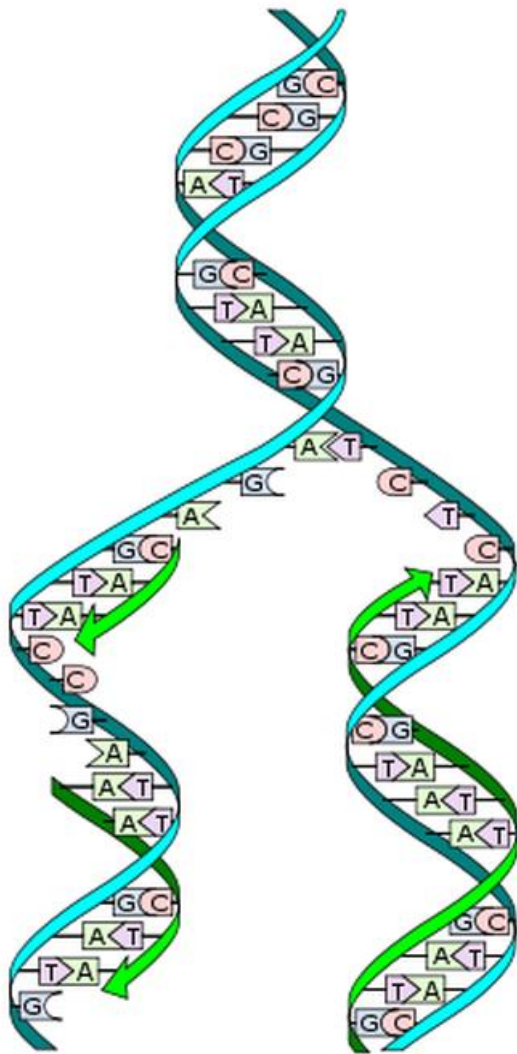


Через 14 ходов мы опять видим обычную шахматную доску.

ДНК

ДНК - дезоксирибонуклеиновая кислота, является главным носителем информации у живых существ. Другая важнейшая функция ДНК – сохранение наследственной информации в результате деления.

Обратимся к википедии:



Процесс репликации: раскручивание двойной спирали ДНК — синтез комплементарных цепей ДНК-полимеразой — образование двух молекул ДНК из одной.

Деление ДНК происходит постепенно, в несколько этапов, причём на выходе образуются две молекулы, обладающие той же информацией, которой владела родительская ДНК.

Давайте сравним это с тем, как происходит деление паттерна в алгоритме Живые Пиксели.

В следующем случае записано число 19 в двоичном виде: 1011. Пиксели постепенно "делятся" подобно ДНК - через четыре шага видим две новые цифры 1011.



Начальное состояние, записано число 19: 1011.



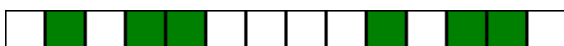
Первый шаг деления



Второй шаг деления – видно, как пиксели «расходятся» в стороны – сравните с рисунком деления ДНК выше.



Третий шаг



Четвёртый шаг – деление окончено, на рисунке два паттерна 1011.

Через 30 шагов изображение возвращается к своему начальному состоянию.

Эволюция

Алгоритм, имитирующий развитие биологических видов - от простейших микроорганизмов до огромных динозавров и разумных людей. Данный алгоритм был разработан до «Живых Пикселей» и является самостоятельным клеточным автоматом.

Правила

Алгоритм Эволюция основан на следующих определениях и правилах.

Элемент — набор пикселей, лежащих рядом. Элементы показаны различными геометрическими фигурами, в зависимости от количества пикселей в них. Столкновение — ситуация, когда пиксели из разных элементов находятся в непосредственной близости. Элементы не могут оставаться друг рядом с другом и столкновение должно быть сразу же исправлено с помощью специальных правил.

Правила алгоритма Эволюция:

1. За один ход добавляется один элемент в один пиксель в случайное место на поле.
2. При столкновении элементов с одинаковым числом пикселей они образуют новый элемент, на один пиксель больше.

В природе — появление более крупного организма при достаточном количестве более мелких

3. При столкновении элементов разного размера, остаётся лишь самый большой

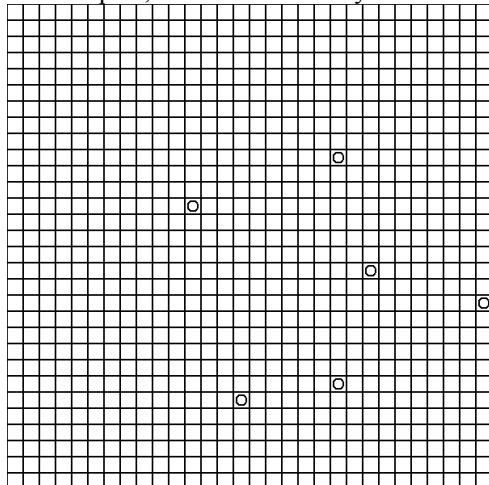
В природе — борьба за выживание, пищевая цепочка

4. Существование элементов ограничено по времени, которое прямо пропорционально количеству пикселей в элементе.

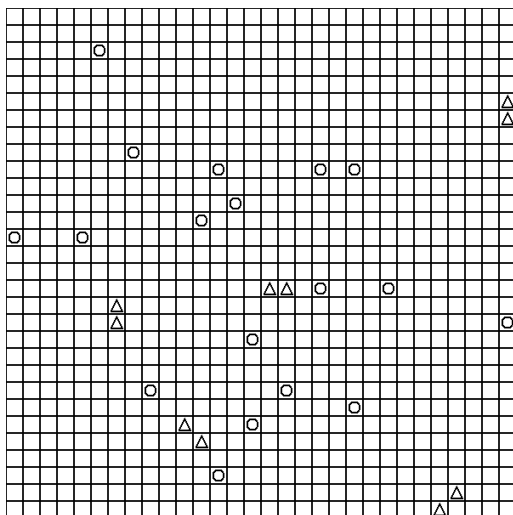
В природе - время жизни организмов ограничено и также есть зависимость времени жизни от среднего размера особи в виде.

Изучая «Эволюцию», можно увидеть в ней развитие отдельных живых существ, при этом правильнее рассматривать её как моделирование развития различных видов животных. Каждый элемент — есть определённый вид организмов.

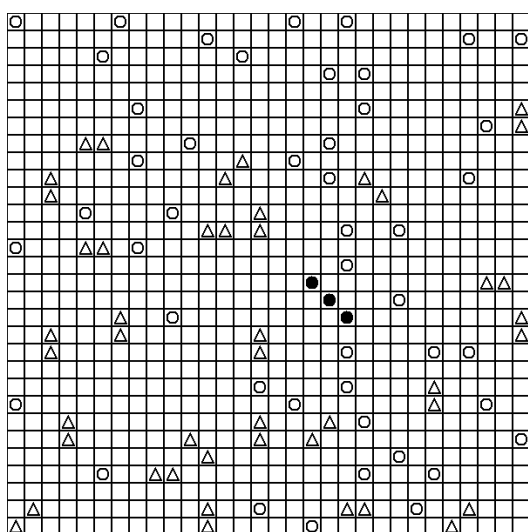
Рассмотрим, как постепенно пустое поле программы наполняется «живыми существами»:



Вначале появляются мелкие организмы в один пиксель, на изображении показаны кружками.

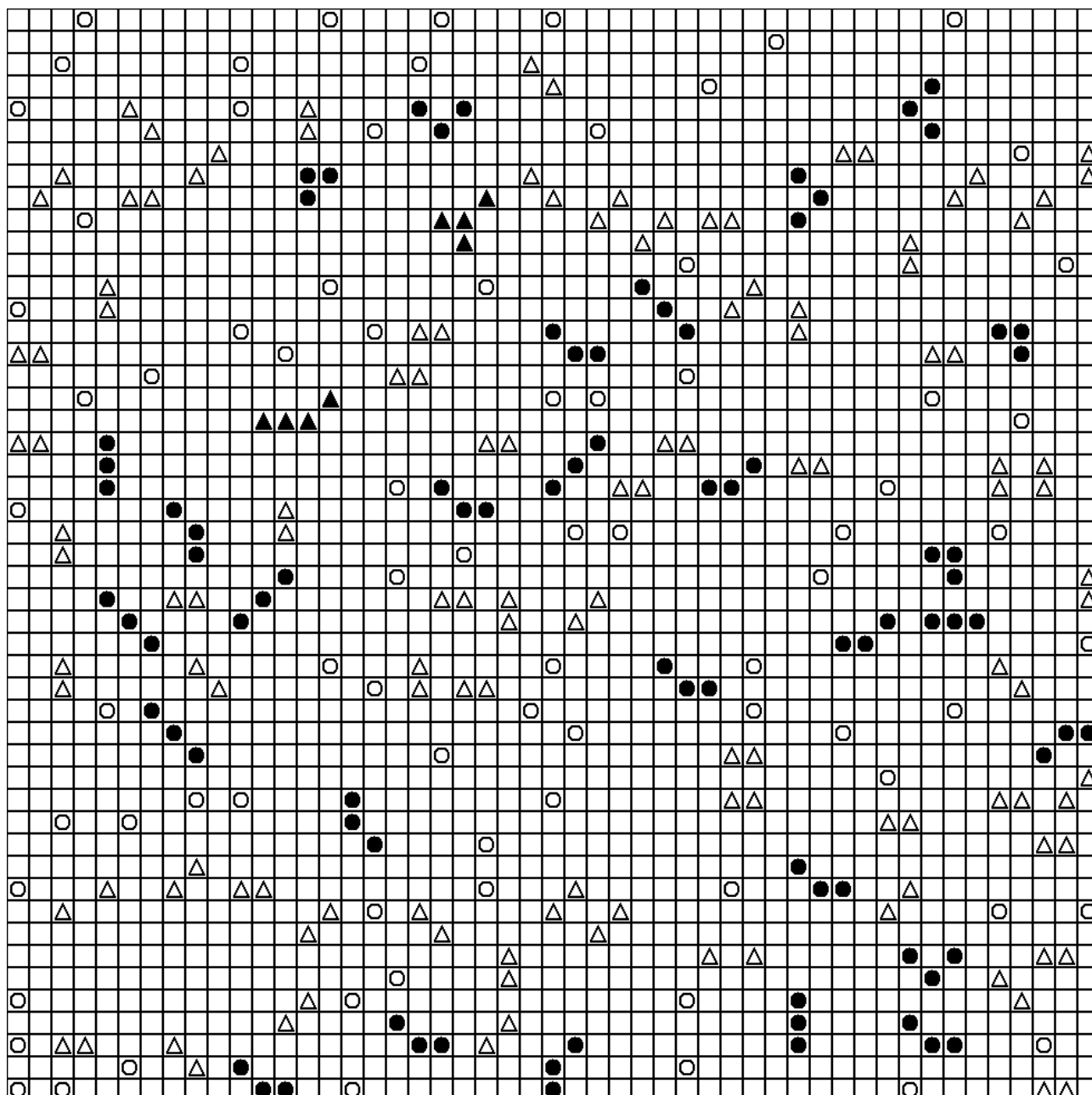


Затем более крупные – в два пикселя, на изображении показаны треугольниками.



Когда поле достаточно насыщено организмами в два пикселя, появляется первый организм в три пикселя – закрашенные кружки в центре. Опять мы видим закон перехода количества в качество.

Для удобства элементы из одного пикселя показаны прозрачными кружками, из двух — прозрачными треугольниками, из трёх — закрашенными кружками, из четырёх — закрашенными треугольниками. Наиболее распространёнными являются элементы из одного и двух пикселей.



На рисунке показана эволюция 2779 шагов, при этом элементов из четырёх пикселей всего два. Наиболее крупные элементы живут долго, затем надолго исчезают — как динозавры.

Статистика видов по размерам из вышеприведённого рисунка:

1 пиксель	2 пикселя	3 пикселя	4 пикселя
78	64	28	2

Эксперименты показывают, что количество элементов в один и два пикселя при большом количестве шагов примерно одинаковое, в данном случае больше элементов из одного пикселя, в некоторых случаях небольшой перевес у треугольников — элементов в два пикселя. Резкое уменьшение количества видов происходит, начиная с величины в три пикселя.

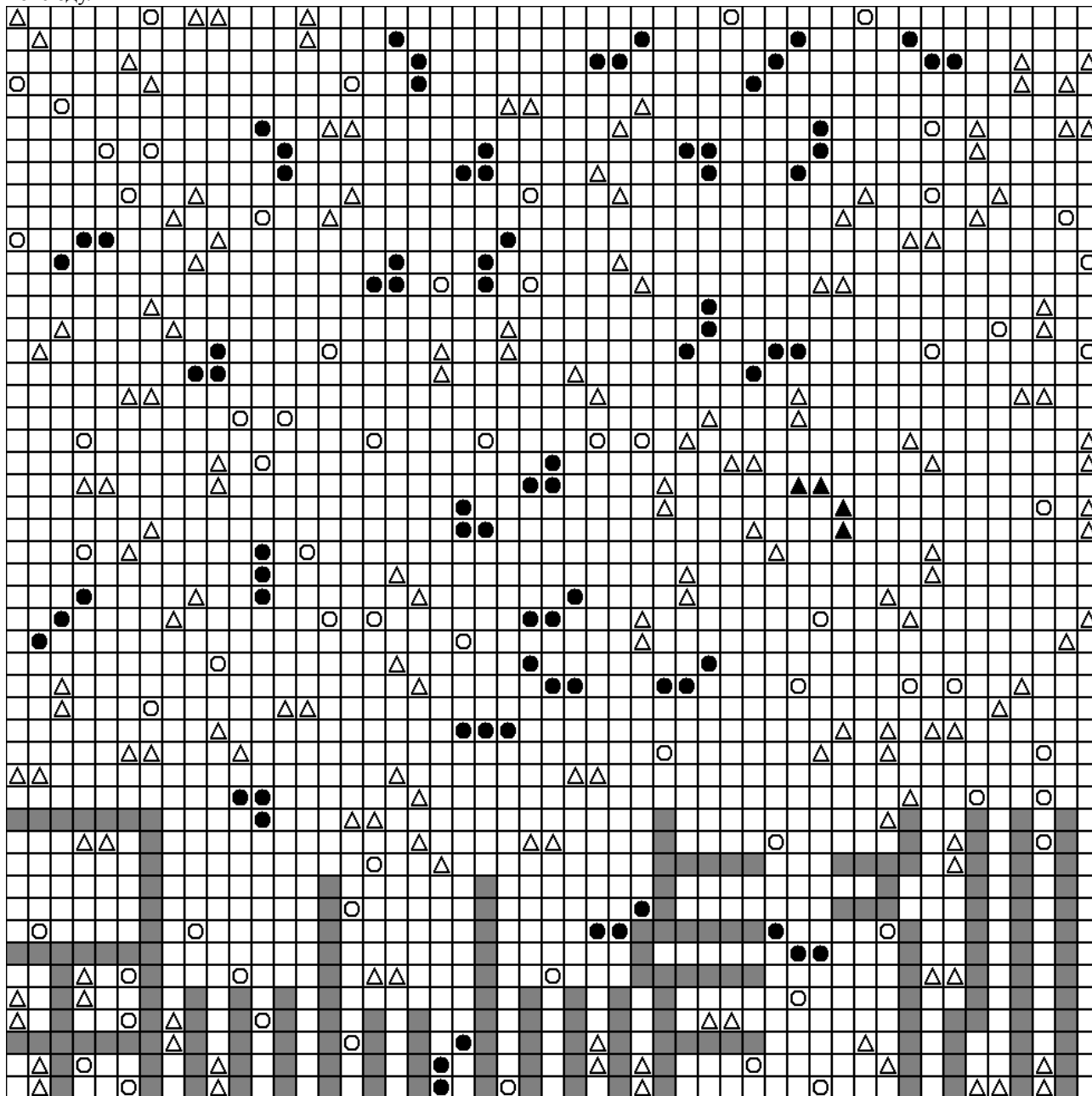
Статистика видов в живой природе:

микроорганизмы: бактерии, вирусы	насекомые	позвоночные	млекопитающие
много	очень много	мало	очень мало

Таким образом, сходство с живой природой налицо. Наиболее крупными животными являются млекопитающие — киты в океане и слоны на суше.

Море

Жизнь вышла из океана. На изображении серые полосы внизу имитируют изрезанное морское дно. Интересно проследить, как зависит размер животных видов от среды обитания — чем больше у вида пространства, тем больше размер наибольших живых существ. При этом мелкие животные встречаются повсюду.



Статистика видов по размерам из вышеприведённого рисунка:

1 пиксель	2 пикселя	3 пикселя	4 пикселя
65	77	26	1

Видна явная связь между размерами вида и местом обитания. В расщелины внизу может "залезть" любой «зверь», при этом в основном там объекты в один или два пикселя. Эта зависимость хорошо известна рыболовам — размер рыбы, например, окуня, сильно зависит от размера водоёма — чем больше водоём, тем больше рыба.

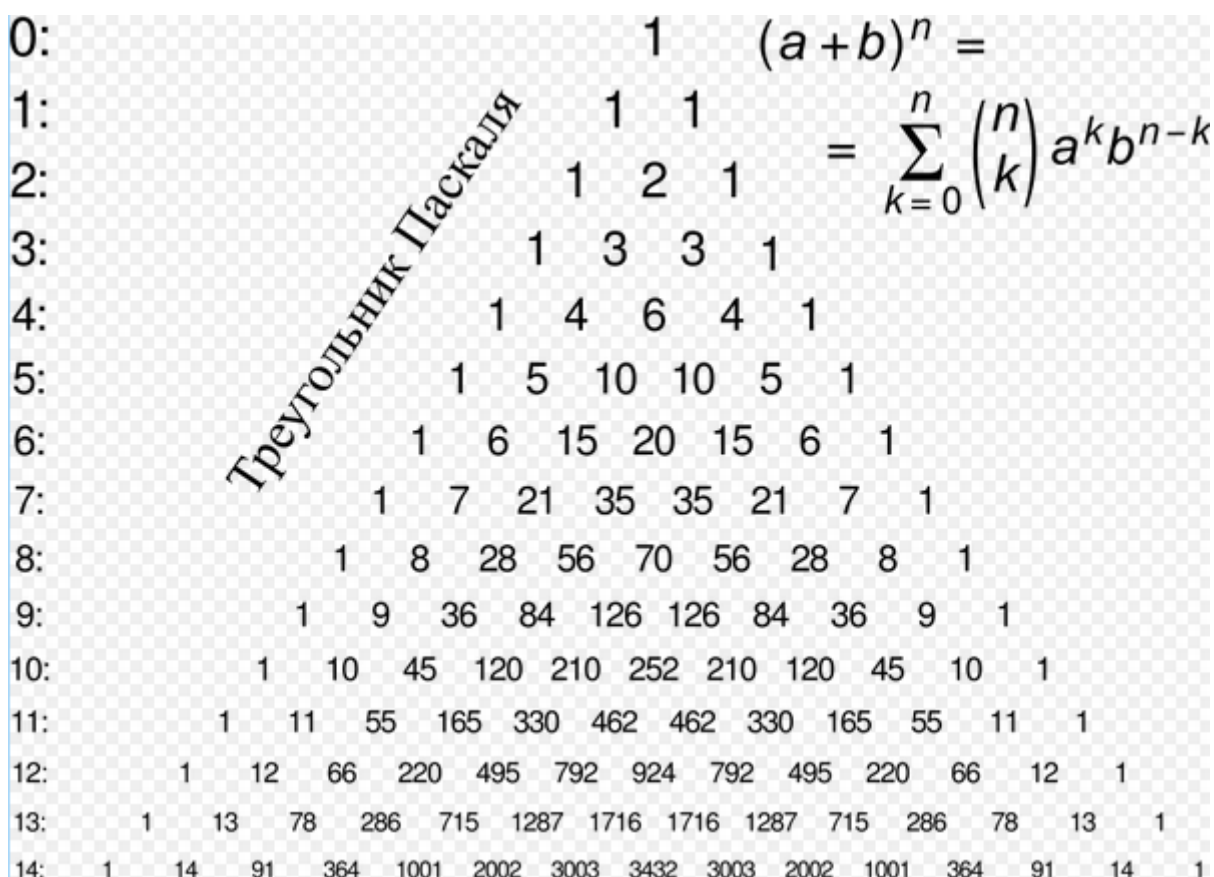
Применение

Алгоритм Эволюция соответствует научному направлению «Искусственная Жизнь». Определение «Искусственной Жизни» из Wikipedia: «Данная область науки изучает механизм процессов, присущих всем живым системам, невзирая на их природу. Хотя этот термин чаще всего применяется к компьютерному моделированию жизненных процессов, он также подходит и к жизни в пробирке (англ. wet alife), изучению

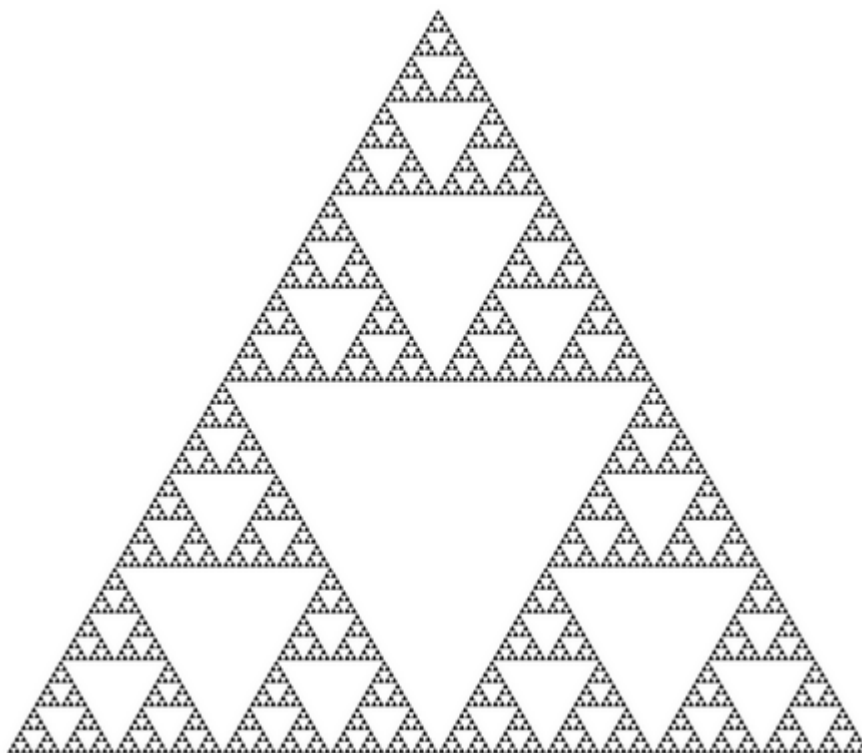
Алгоритм Эволюция может быть использован в качестве базовой программной абстракции для изучения процессов эволюции в живой природе. Простые и очевидные начальные правила и наглядное графическое представление делают его очень удобным для этой цели. С другой стороны, алгоритм может быть использован в направлении развития искусственного интеллекта, где уже есть успешный опыт применения «генетического алгоритма», базирующегося на теории эволюции Дарвина, который также иногда называют «эволюционным».

Проброобразом современных клеточных автоматов можно считать так называемый треугольник Паскаля. Впервые треугольник описал древнеиндийский математик Пингала во II веке до нашей эры. Вот что о треугольнике пишет Википедия:

“Если очертить треугольник Паскаля, то получится равнобедренный треугольник. В этом треугольнике на вершине и по бокам стоят единицы. Каждое число равно сумме двух расположенных над ним чисел. Продолжать треугольник можно бесконечно. Строки треугольника симметричны относительно вертикальной оси. Имеет применение в теории вероятностей и обладает занимательными свойствами.”



Одно из свойств треугольника Паскаля читатель может проверить самостоятельно: Каждое число в треугольнике равно количеству способов добраться до него из вершины, перемещаясь либо вправо-вниз, либо влево-вниз. Из треугольника Паскаля выводятся простые числа, числа Фибоначчи, числа Каталана и другие важные математические последовательности. Если заменить нечётные числа точками, а чётные сделать прозрачными, то получаем треугольник Серпинского:



Рекомендую обзор треугольников Паскаля и Серпинского, опубликованный на Арбузе [7].

Клеточный Автомат

Опять обратимся к Википедии [1]: "Станислав Улам, работая в Лос-аламосской национальной лаборатории в 1940-е годы, изучал рост кристаллов, используя простую решёточную модель. В это же время Джон фон Нейман, коллега Улама, работал над проблемой самовоспроизводящихся систем. Первоначальная концепция фон Неймана основывалась на идее робота, собирающего другого робота. Такая модель известна как кинематическая. Разработав эту модель, фон Нейман осознал сложность создания самовоспроизводящегося робота и, в частности, обеспечения необходимого "запаса частей", из которого должен строиться робот. Улам предложил фон Нейману использовать более абстрактную математическую модель, подобную той, что Улам использовал для изучения роста кристаллов. Таким образом, возникла первая клеточно-автоматная система. Подобно решётке Улама, клеточный автомат фон Неймана двумерный, а самовоспроизводящийся робот описан алгоритмически. Результатом явился универсальный конструктор, работающий "внутри" клеточного автомата с окрестностью, включающей непосредственно прилегающие ячейки, и имеющего 29 состояний. Фон Нейман доказал, что для такой модели существует паттерн, который будет бесконечно копировать самого себя"

Алгоритм Живые Пиксели на бесконечном поле бесконечно копирует любое начальное расположение активных пикселей, то есть решает задачу фон Неймана.

Определение клеточного автомата из Википедии:

"Клёточный автомат — дискретная модель, изучаемая в математике, теории вычислимости, физике, теоретической биологии и микромеханике. Включает регулярную решётку ячеек, каждая из которых может находиться в одном из конечного множества состояний, таких как 1 и 0. Решётка может быть любой размерности. Для каждой ячейки определено множество ячеек, называемых соседством. К примеру, соседство может быть определено как все ячейки на расстоянии не более 2 от текущей. Для работы клеточного автомата требуется задание начального состояния всех ячеек, и правил перехода ячеек из одного состояния в другое. На каждой итерации, используя правила перехода и состояния соседних ячеек, определяется новое состояние каждой ячейки. Обычно правила перехода одинаковы для всех ячеек и применяются сразу ко всей решётке"

Достаточно обширный анализ клеточных автоматов сделан в программе Cellular Automata Laboratory <http://www.fourmilab.ch/cellab/cellab.zip> В программе реализованы 37 видов автоматов, которые описаны по-английски на странице: <http://www.fourmilab.ch/cellab/manual/rules.html> Примеры к программе не используют алгоритм, повторяющий Живые Пиксели, изучение описаний алгоритмов показывает, что Живые Пиксели являются частным случаем алгоритма Эдварда Фредкина Fredmem. Близким к Живым Пикселям является алгоритм Fractal, который также является частным случаем алгоритма Fredmem. Эдвард Фредкин указывает

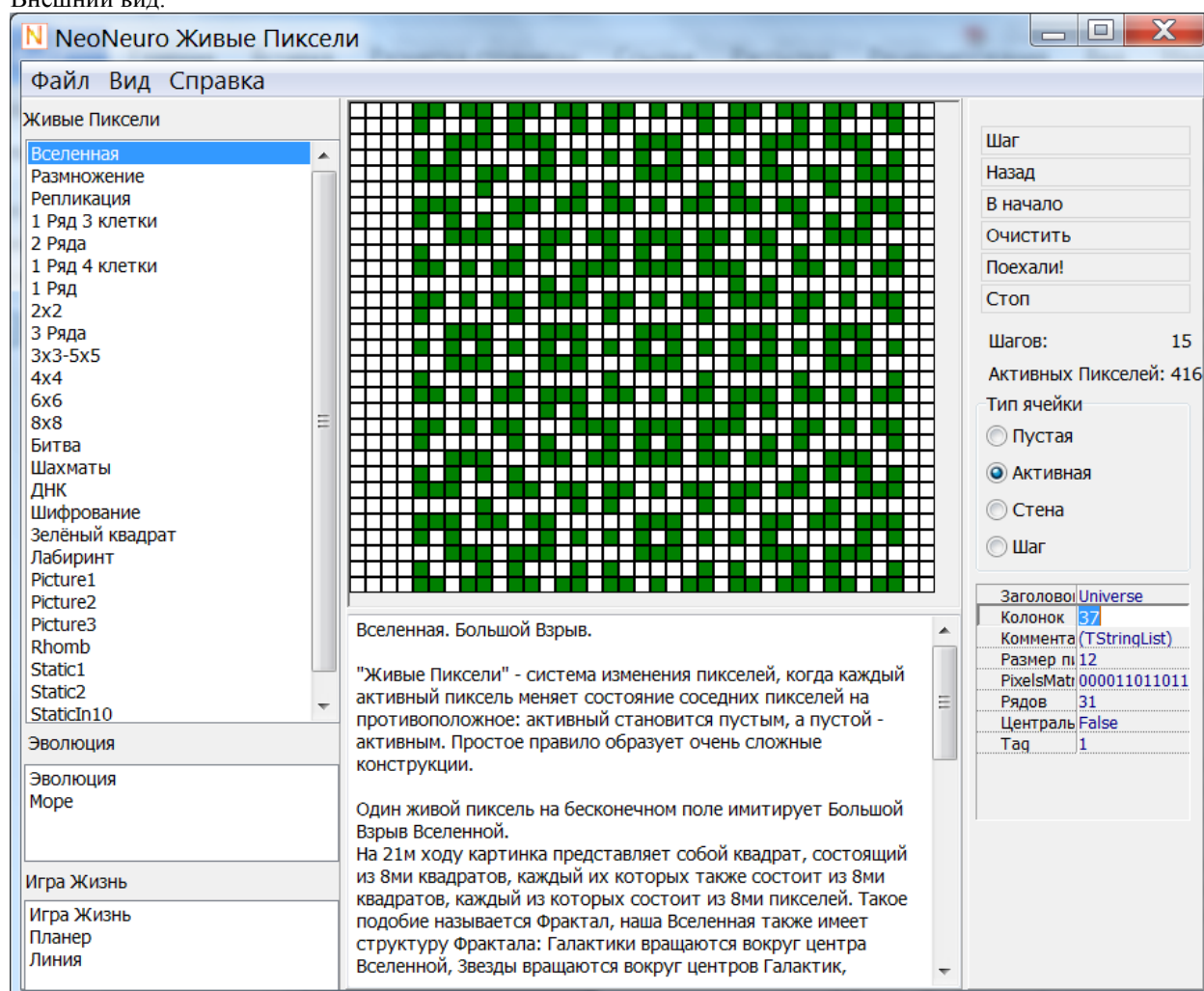
на то, что его алгоритм является примером «самовоспроизводства», то есть «размножения» - начальная фигура через определённое число итераций превращается в несколько таких же фигур. Такую картину мы наблюдаем в Живых Пикселях на бесконечном поле.

Ещё более полная подборка клеточных автоматов приводится на сайте: http://www.mirekw.com/ca/rullex_wlif.html

Алгоритмы «Эволюция», и затем «Живые Пиксели» были разработаны и программно реализованы в 2012м году. На сегодняшний день многие аспекты алгоритмов остаются неизученными. Читатель может принять участие в анализе «Живых Пикселей» и «Эволюции» и поиске интересных конструкций, используя программу NeoNeuro Live Pixels – Живые Пиксели.

Программа NeoNeuro Живые Пиксели

Внешний вид:



Программа показывает поле с «пикселями» и описание каждого алгоритма. Простой и удобный редактор позволяет создавать новые конструкции, анализировать их работу и сохранять в файл, который можно выложить в интернет для всеобщего обозрения.

Левая панель

Позволяет выбрать сцену для любого из трёх алгоритмов:

Живые Пиксели

Эволюция

Игра Жизнь

Центральная панель

Поле развития клеточных алгоритмов. Кликом мыши можно сделать ход. "Тип ячейки" в правой панели позволяет менять функциональность клика мыши.

Внизу центральной панели - описание конкретного алгоритма.

Правая панель

Шаг - один шаг программы. Равносильно клику мыши в режиме "шаг".

Назад - возвращает на один шаг назад.

Очистить - обнуляет поле.

В начало - возвращает поле в начальное положение.

Поехали! - автоматически меняет сцены. Кнопка превращается в "Быстрее".

Стоп - остановка автоматической смены сцен.

Тип ячейки - выбор функционала для клика мыши по матрице. Движение мышью с нажатой левой клавишей по матрице позволяет "рисовать" требуемое изображение быстрее.

В нижней части правой колонки расположены основные элементы настройки матрицы, данные параметры можно сохранить в файл для последующей работы.

Для настройки доступны:

- Заголовок
- Ширина и высота матрицы
- Размер пикселя (клетки)
- Комментарий – описание паттерна

Интерфейс программы реализован на русском, английском и немецком языках, пользователи имеют возможность добавить дополнительные переводы.

Заключение

Спасибо за то, что ознакомились с миром Живых Пикселей. Надеюсь, он Вам понравился. Приглашаю всех желающих продолжить поиск интересных явлений и конструкций в «Живых Пикселях» и «Эволюции» – редактор в программе прост и удобен для этих целей.

Сайт: www.neoneuro.com

Список использованной литературы

1. http://ru.wikipedia.org/wiki/Клеточный_автомат
2. Обзор клеточных автоматов с примерами: http://www.mirekw.com/ca/rullex_life.html
3. Обзор клеточных автоматов: http://en.wikipedia.org/wiki/Life-like_cellular_automata#A_selection_of_Life-like_rules
4. Эдвард Фредкин: http://en.wikipedia.org/wiki/Edward_Fredkin
5. Треугольник Паскаля http://ru.wikipedia.org/wiki/Треугольник_Паскаля
6. Автомат 2x2 <http://www.conwaylife.com/wiki/index.php?title=2x2>
7. Обзор треугольников Паскаля и Серпинского http://arbuz.uz/u_treug.html