Материал взят с http://inf.1september.ru/view\_article.php?ID=200901509.

**Что наша «Жизнь»? Игра!**

Одной из легенд раннего программирования по праву является игра, придуманная американским математиком Джоном Хортоном Конуэем (другой вариант русского произношения фамилии — Конвей). Вот уже несколько десятков лет она привлекает к себе пристальное внимание. Созданы десятки программ, реализующих эту игру чуть ли не на всех типах компьютеров, написаны тысячи статей, сотни сайтов в Интернете посвящены этой игре.

В свое время она была настолько популярной среди программистов, что съела немало их рабочего времени и машинного времени первых суперкомпьютеров.

Вообще-то большая часть работ Конуэя относится к области чистой математики, но, помимо серьезных исследований, он увлекается также занимательной математикой. Настоящая же статья посвящена самому знаменитому детищу Конуэя — игре, которую сам Конуэй назвал “Жизнь”.

Для игры “Жизнь” вам не понадобится партнер — в нее можно играть и одному. Возникающие в процессе игры ситуации очень похожи на реальные процессы, происходящие при зарождении, развитии и гибели колоний живых организмов. По этой причине “Жизнь” можно отнести к категории так называемых “моделирующих игр” — игр, которые в той или иной степени имитируют процессы, происходящие в реальной жизни. Уже довольно давно никто не играет в эту игру без использования компьютера, хотя это вполне возможно. Для этого понадобилась бы довольно большая доска, разграфленная на клетки, и много плоских фишек двух цветов (например, просто несколько наборов обычных шашек небольшого диаметра или одинаковых пуговиц двух цветов). Можно также воспользоваться доской для игры в го, но тогда вам придется раздобыть маленькие плоские шашки, которые свободно умещаются в ячейках этой доски. (Обычные камни для игры в го не годятся, потому что они не плоские.) Можно также рисовать ходы на бумаге, но значительно проще, особенно для начинающих, играть, переставляя фишки или шашки на доске.

Основная идея игры состоит в том, чтобы, начав с какого-нибудь простого расположения фишек (организмов), расставленных по различным клеткам доски, проследить за эволюцией исходной позиции под действием “генетических законов” Конуэя, которые управляют рождением, гибелью и выживанием фишек. Конуэй тщательно подбирал свои правила и долго проверял их “на практике”, добиваясь, чтобы поведение популяции было достаточно интересным, а главное, непредсказуемым.

Каждая клетка на бесконечном во все стороны поле имеет ровно восемь соседей. Рождаются и погибают клетки по следующим правилам:

**1.** Если фишка имеет четырех или более соседей, то она умирает от перенаселенности (с этой клетки снимается фишка).

**2.** Если фишка не имеет соседей или имеет ровно одного соседа, то она умирает от нехватки общения.

**3.** Если клетка без фишки имеет ровно трех соседей, то в ней происходит рождение (на клетку кладется фишка).

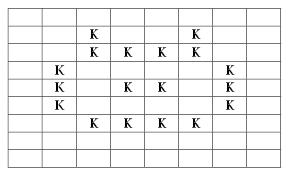
**4.** Если не выполнено ни одно из перечисленных выше условий, состояние клетки не изменяется.

Игра эта пошаговая, и за один шаг игры со всеми клетками поля одновременно происходят (или не происходят) изменения, описанные тремя вышеуказанными правилами.

На бумаге (или на экране) играют в эту игру так: рисуют какую-нибудь колонию закрашенных клеток, а затем шаг за шагом прослеживают ее эволюцию.

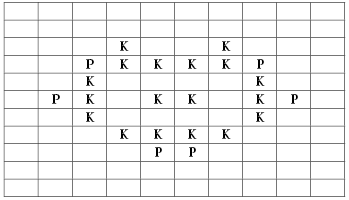
Давайте для начала поучимся играть сами.

Итак, исходная колония клеток выглядит следующим образом (см. *рис*. 1):



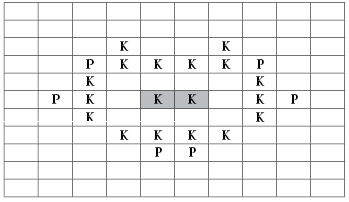
*Рис*. 1. Колония клеток в игре “Жизнь”, носящая название “Чеширский кот”.   
В исходной конфигурации легко просматриваются ушки и глазки

Давайте теперь буквами “Р” отметим те пустые клетки, на которых произойдет рождение новых закрашенных клеток. Напомним, что это такие пустые клетки, которые имеют ровно трех закрашенных соседей:



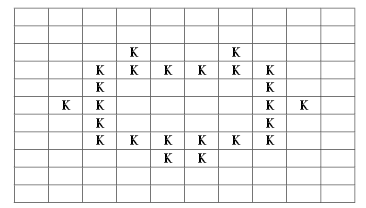
*Рис*. 2. Колония клеток с отмеченными “новорожденными” клетками

Теперь заштрихуем те старые клетки, которые должны погибнуть. Напомним, что это такие закрашенные плитки, которые имеют либо меньше двух закрашенных соседей, либо больше трех. При расчетах, разумеется, не надо принимать во внимание клеточки, отмеченные буквой “Р”, — на них процесс рождения еще не закончился:



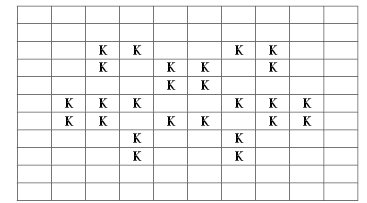
*Рис*. 3. Колония клеток с отмеченными клетками, которые должны погибнуть

И, наконец, уберем с заштрихованных полей плитки, а поля, отмеченные буквой “Р”, отметим буквой “К”:

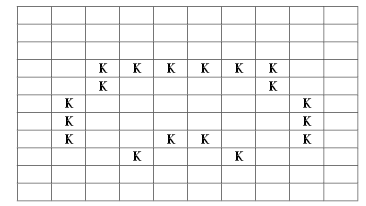


*Рис*. 4. Колония клеток “Чеширский кот” на втором шаге игры. Глазки закрылись, зато морда стала шире, и это уже точно кот, а не кошечка

Повторяя этот процесс cнова и снова, мы должны получить следующие конфигурации:



*Рис.* 5. “Чеширский кот” на третьем шаге. Появился во всей красе, только ушки слегка обвисли



*Рис*. 6. “Чеширский кот” на четвертом шаге игры. Морда с усиками

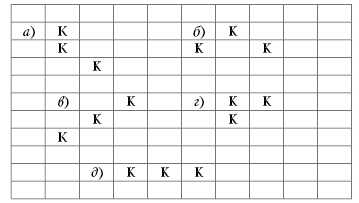
Надеемся, что для вас не составит труда проследить эволюцию этой колонии клеток до самого конца.

Добавим, что конфигурация, по которой мы учились играть, была открыта К.Р. Томпкинсом из Короны, штат Калифорния.

Начав игру, вы сразу заметите, что популяция непрестанно претерпевает необычные, нередко очень красивые и всегда неожиданные изменения. Иногда первоначальная колония организмов постепенно вымирает, т.е. все фишки исчезают, однако произойти это может не сразу, а лишь после того, как сменится очень много поколений. В большинстве своем исходные конфигурации либо переходят в устойчивые (последние Конуэй называет “любителями спокойной жизни”) и перестают изменяться, либо навсегда переходят в колебательный режим. При этом конфигурации, не обладавшие в начале игры симметрией, обнаруживают тенденцию к переходу в симметричные формы. Обретенные свойства симметрии в процессе дальнейшей эволюции не утрачиваются, а симметрия конфигурации может лишь обогащаться.

После первых, достаточно бессистемных, попыток проследить эволюцию какой-либо колонии можно перейти к более осмысленным действиям, продемонстрировав ученикам основы научного мышления.

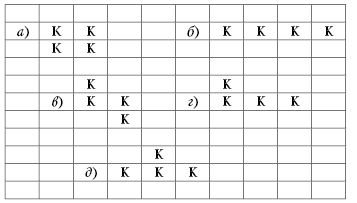
Можно, скажем, проследить эволюцию всех возможных “триплетов” — комбинаций из трех клеток. Выживает триплет лишь в том случае, если по крайней мере одна фишка граничит с двумя занятыми клетками. Пять триплетов, не исчезающих на первом же ходу, изображены на *рис*. 7. (При этом ориентация триплетов, т.е. как они расположены на плоскости — прямо, “вверх ногами” или косо, не играет никакой роли.) Первые три конфигурации (*а*, *б*, *в*) на втором ходу погибают. Относительно конфигурации *в* заметим, что любой диагональный ряд фишек, каким бы длинным он ни оказался, с каждым ходом теряет стоящие на его концах фишки и в конце концов совсем исчезает. Скорость, с которой шахматный король перемещается по доске в любом направлении, Конуэй называет “скоростью света”. Пользуясь этой терминологией, можно сказать, что любой диагональный ряд фишек распадается с концов со скоростью света.



*Рис*. 7. Эволюция пяти триплетов

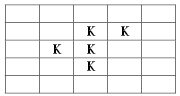
Конфигурация *г* на втором ходу превращается в устойчивую конфигурацию — “блок” (квадрат размером 2  2). Конфигурация *д* служит простейшим примером так называемых “флип-флопов” (кувыркающихся конфигураций, возвращающихся в исходное состояние через каждые два хода). При этом она попеременно превращается то в вертикальный, то в горизонтальный ряд из трех фишек. Конуэй называет этот триплет “мигалкой”.

На *рис*. 8 изображены пять тетрамино (четыре клетки, из которых состоит элемент тетрамино, связаны между собой ходом ладьи). Как вы уже видели (если довели до конца эволюцию “Чеширского кота”), квадрат *а* относится к категории “любителей спокойной жизни”. Конфигурации *б* и *в* после второго хода превращаются в устойчивую конфигурацию, называемую “ульем”. Отметим попутно, что “ульи” возникают в процессе игры довольно часто. Тетрамино, обозначенное буквой *г*, также превращается в улей, но на третьем ходу. Особый интерес представляет тетрамино *д*, которое после девятого хода распадается на четыре отдельные “мигалки”. Вся конфигурация носит название “навигационные огни”, или “светофоры”. “Светофоры” относятся к разряду флип-флопов и возникают в игре довольно часто.



*Рис*. 8. Пять видов тетрамино

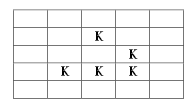
Предоставляем самостоятельно поэкспериментировать на досуге с двенадцатью фигурами пентамино (фигуры, состоящие из пяти фишек, связанных между собой так, что их клетки можно обойти ходом ладьи) и посмотреть, во что они превращаются. Оказывается, что пять из них на пятом ходу погибают, две быстро переходят в устойчивые конфигурации из семи фишек, а четыре после небольшого числа ходов превращаются в “навигационные огни”. Единственным исключением в этом смысле является элемент пентамино, имеющий форму буквы r (*рис*. 9), превращения которого заканчиваются не столь быстро (превращения конфигурации считаются исчерпанными, если та исчезает, переходит в устойчивую конфигурацию или начинает периодически пульсировать).



*Рис*. 9. r-пентамино

Конуэй проследил развитие r-образного пентамино вплоть до четыреста шестидесятого хода, после которого данная конфигурация распалась на множество “глайдеров”. Конуэй пишет, что “от фигуры осталось множество мертвых (не изменяющихся) обломков и лишь несколько малых областей, в которых все еще теплилась жизнь, так что отнюдь не очевидно, что процесс эволюции должен происходить бесконечно долго”.

Одним из самых замечательных открытий Конуэя следует считать конфигурацию из пяти фишек под названием “глайдер”, изображенную на *рис*. 10. После второго хода “глайдер” немного сдвигается и отражается относительно диагонали. В геометрии такой тип симметрии называется “скользящим отражением”, отсюда же и происходит название фигуры.   
В результате двух последующих ходов “глайдер” “выходит из пике”, ложится на прежний курс и сдвигается на одну клетку вправо и на одну клетку вниз относительно начальной позиции. Выше уже отмечалось, что скорость шахматного короля в игре “Жизнь” принято называть скоростью света. Выбор Конуэя пал именно на этот термин из-за того, что в изображенной им игре большие скорости просто не достигаются. Ни одна конфигурация не воспроизводит себя достаточно быстро, чтобы двигаться с подобной скоростью. Конуэй также доказал, что максимальная скорость по диагонали составляет одну четверть скорости света. Поскольку “глайдер” воспроизводит сам себя после четырех ходов и при этом опускается на одну клетку по диагонали, то говорят, что он скользит по полю со скоростью, равной одной четвертой скорости света.



*Рис*. 10. “Глайдер?”

Конуэй также показал, что скорость любой конечной фигуры, перемещающейся по вертикали или по горизонтали на свободные клетки, не может превышать половину скорости света. Сумеете ли самостоятельно найти достаточно простую фигуру, которая движется с такой скоростью? Напомним, что скорость движения определяется дробью, в знаменателе которой стоит число ходов, необходимых для воспроизведения фигуры, а в числителе — число клеток, на которое она при этом смещается. Например, если какая-нибудь фигура за каждые четыре хода передвигается на две клетки по вертикали или по горизонтали, повторяя свою форму и ориентацию, то скорость такой фигуры будет равна половине скорости света. Надо сказать, что поиски перемещающихся по доске фигур — дело чрезвычайно сложное. Конуэю известны всего четыре такие конфигурации, которые он называет “космическими кораблями”. В их число входит уже известный нам “глайдер”. (“Глайдер” считается “космическим кораблем” легчайшего веса, потому что все остальные корабли состоят из большего числа фишек.)

Конуэй исследовал эволюцию всех горизонтальных рядов из *N* фишек вплоть до *N* = 20. Мы уже знаем, что происходит при *N* http://inf.1september.ru/2009/15/LesQ.gif4. Ряд из пяти фишек переходит в “навигационные огни”, ряд из шести фишек исчезает, из семи фишек получается “пасека”, из восьми — четыре “улья” и четыре “блока”, девять фишек превращаются в два комплекта “навигационных огней”, а ряд, состоящий из десяти фишек, переходит в “пентадекатлон” — периодически воспроизводящую себя конфигурацию с периодом, равным 15. Ряд из одиннадцати фишек эволюционирует, превращаясь в две “мигалки”; двенадцать фишек в конце концов переходят в два “улья”, а тринадцать — снова в две “мигалки”. Если ряд состоит из 14 или 15 фишек, то он полностью исчезает, а если фишек 16, то получается большой набор “навигационных огней”, состоящий из восьми “мигалок”. Эволюция ряда из 17 фишек завершается возникновением четырех “блоков”; ряды, состоящие из 18 или 19 фишек, также полностью исчезают с доски, и, наконец, эволюция ряда из 20 фишек завершается появлением двух “блоков”.

**Реализация игры “Жизнь”**

***Жизнь в стиле КуМир (Школьный алгоритмический язык)***

Игровое поле с фишками будем моделировать в виде квадратного двумерного массива с элементами символьного типа. Имя этого массива — поле, а размер — размер. На поле и в соответствующем массиве фишки будем изображать в виде символа “\*”.

Для подсчета числа соседних фишек для клетки с координатами (i, j) создадим функцию СчетчикСоседей:

**алг цел** СчетчикСоседей(**арг цел** i, j)

**нач цел** счетчик, ii, jj

счетчик := 0

**нц для** ii **от** i — 1 до i + 1

**нц для** jj от j — 1 до j + 1

**если** поле[ii, jj] = "\*"

**то**

счетчик := счетчик + 1

**все**

**кц**

**кц**

|Исключаем из подсчета фишку в клетке с координатами (i, j)

**если** поле[i, j] = "\*"

**то**

счетчик := счетчик — 1

**все**

**знач** := счетчик |Значение функции

**кон**

Значения числа соседних фишек для всех клеток игрового поля будем хранить в массиве с именем ВсегоСоседей.

Примем, что в крайних строках и столбцах массива (поля) фишки находиться не могут, поэтому процедура заполнения этого массива будет выглядеть так:

**алг** ЗаполнениеМассиваВсегоСоседей

**нач цел** i, j

**нц для** i **от** 2 **до** размер — 1

**нц для** j **от** 2 **до** размер — 1

ВсегоСоседей[i, j] := СчетчикСоседей(i, j)

**кц**

**кц**

**кон**

По значениям элементов массива ВсегоСоседей можно сформировать новое поколение фишек (новую ситуацию на игровом поле). Соответствующая процедура имеет вид:

**алг** НовоеПоколение

**нач цел** i, j, всего

|Заполняем массив ВсегоСоседей

ЗаполнениеМассиваВсегоСоседей

|Меняем ситуацию на игровом поле

**нц для** i **от** 2 **до** размер — 1

**нц для** j **от** 2 **до** размер — 1

|Чтобы многократно не использовать значение ВсегоСоседей[i, j],

|применим величину "всего"

всего := ВсегоСоседей[i, j]

**если** всего = 0 **или** всего = 1 **или** всего > 3

**то** |Гибель

поле[i, j] := " "

**все**

**если** всего = 3

**то** |Рождение или выживание

поле[i, j] := "\*"

**все**

**кц**

**кц**

ВыводПоколения

**кон**

— где ВыводПоколения — процедура вывода на экран всех элементов массива поле:

**алг** ВыводПоколения

**нач цел** i, j

**нц для** i **от** 1 **до** размер

**нц для** j **от** 1 **до** размер

**вывод** поле[i, j]

**кц**

**вывод** **нс**

**кц**

**кон**

Надо продумать, как задавать исходную ситуацию.

**Постановка задачи.**

Дана реализация игры «Жизнь». Здесь пользователь не может задавать первоначальную ситуацию. Сделайте так, чтобы пользователь смог задать первоначальную ситуацию.

Файл htm.

<!DOCTYPE html>

<html>

<head>

<TITLE>Жизнь</TITLE>

<script src ="main.js"> </script>

</head>

<body onload=drawTable()>

<H1>Игра Жизнь</H1>

<FORM>

<INPUT id="start" type="button" value="Начало"

onclick="Nachalo()">

<INPUT id="clear" type="button" value="Продолжить"

onclick="NewGeneration()">

</FORM>

</body>

</html>

Файл main.js.

height=10;

width=10;

field=new Array();

TotalNeighbours=new Array();

function drawTable() {

var element=document.createElement ("table");

element.setAttribute("border","1")

element.setAttribute("id","tab\_life")

for (var i=0; i<height; i++) {

var row=element.insertRow(i);

for (var j=0; j<width; j++) {

var cell=row.insertCell(j);

cell.width="10";

cell.height="10";

cell["id"]="c"+i+"\_"+j;

}

}

document.body.appendChild(element);

}

function CountNeighbours(i, j){

var count =0;

for (var ii =i - 1;ii<=i + 1; ii++)

for (var jj = j - 1; jj<= j + 1; jj++)

if (field[ii][jj]=='\*')

count ++;

//{Исключаем из подсчета фишку в клетке с координатами (i, j)}

if (field[i][j] == '\*')

count--;

return count;

}

function FillArrayTotalNeighbours(){

for (var i = 1; i< height - 1; i++)

for (var j= 1; j<width - 1; j++)

TotalNeighbours[i][j] =CountNeighbours(i, j)

}

function NewGeneration() {

var total;

FillArrayTotalNeighbours();

for (var i = 1; i< height - 1; i++)

for (var j= 1; j<width - 1; j++) {

total= TotalNeighbours[i][j];

if ((total ==0) || (total == 1) || (total > 3))

field[i][j] = ' ';

if (total == 3)

field[i][j]= '\*';

}

PrintGeneration();

}

function PrintGeneration() {

var id, st;

for (var i = 0; i< height; i++){

for (var j= 0; j<width; j++){

id = "c"+i+"\_"+j;

st = document.getElementById(id).style;

if (field[i][j]=='\*')

st.backgroundColor="#f00";

else

st.backgroundColor="#fff";

}

}

}

function Nachalo(){

for (var i = 0; i< height; i++){

field[i]=new Array();

TotalNeighbours[i]=new Array();

for (var j= 0; j<width; j++){

field[i][j]='';

TotalNeighbours[i][j]=0;

}

}

field[1][2]='\*';

field[2][3]='\*';

field[3][1]='\*';

field[3][2]='\*';

field[3][3]='\*';

PrintGeneration()

}