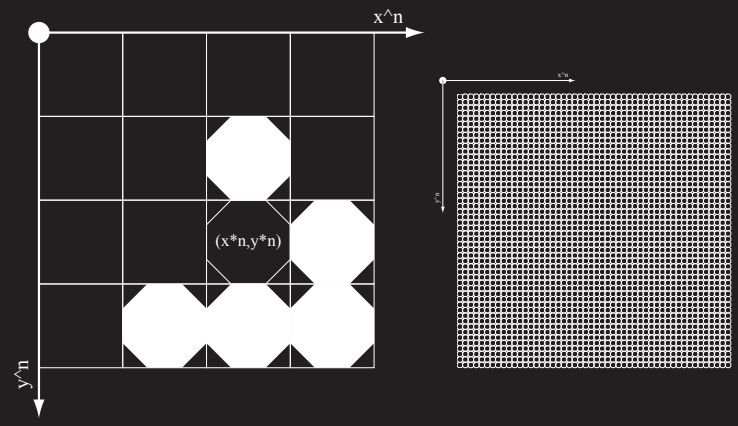
conway´s game of life

“Das Spiel des Lebens (engl. Conway’s Game of Life) ist ein vom Mathematiker John Horton Conway 1970 entworfenes System, basierend auf einem zweidimensionalen zellulären Automaten. Es ist eine einfache und bis heute populäre Umsetzung der Automaten-Theorie von Stanisław Marcin Ulam.” (http://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens; stand: 13.11.10)

Das Spiel besteht aus einem theoretisch unendlichem Spielfeld (Zeilen und Spalten); jedes Feld dieser Matrix ist eine Zelle (Zellularer Automat) welche je nach Systemdurchgang genau zwei Zustände einnehmen kann; nämlich tod bzw. lebend (in meinem Beispiel sichtbar bzw. unsichtbar)

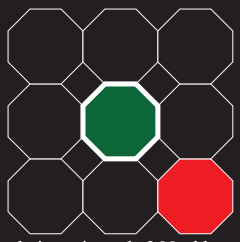


Diese Zustandsänderung je nach Systemdurchgang ergibt sich durch vier einfache Regeln; diese vier Regeln beziehen sich immer direkt auf die Zelle bzw. das Feld; man könnte auch sagen die Reglementierung verhält sich im Bezug zur Zelle statisch; die Dynamik entsteht erst im Bezug zu den acht Nachbarn jeder einzelnen Zelle - unabhängig davon ob die Zelle “tot” oder “lebend” ist;

Die 4 Regeln Conways:
//
01 Eine tote Zelle mit genau drei lebenden Nachbarn wird in der Folgegeneration neu geboren.
02 Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben in der Folgegeneration an Einsamkeit.
03 Eine lebende Zelle mit zwei oder drei lebenden Nachbarn bleibt in der Folgegeneration lebend.
04 Lebende Zellen mit mehr als drei lebenden Nachbarn sterben in der Folgegeneration an Überbevölkerung.

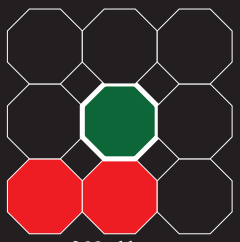
(die 4 Regeln aus: http://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens; stand: 13.11.10)

Die lebende Zelle in einer 23/3 Welt:
STIRBT bei:



_bei weniger als 2 Nachbarn

LEBT bei:



_genau 2 Nachbarn

_mehr als 3 Nachbarn

_genau 3 Nachbarn

Die tote Zelle in einer 23/3 Welt:
LEBT/ bzw. wird geboren bei:

_genau 3 Nachbarn

d.h. es ist erforderlich die Zellen einer grundlegenden Klassifikation zu unterziehen; da sind einerseits:
- Tote Zellen bzw. zur Geburt freistehende Zellen;
- Lebende Zellen denen der Tod bevor steht;

dies wiederum bedeutet das am Ende nur drei effektive Reglementierungen für die zwei Zellzustände übrig bleiben:

[Die tote Zelle]

- hat sie genau drei Nachbarn wird sie leben;(Nachbar=2 --> Zelle wird leben)

[Die lebende Zelle]

- weniger als zwei Nachbarn; die Zelle stirbt; (Nachbar<2 --> Zelle stirbt)
- mehr als drei Nachbarn; die Zelle stirbt; (Nachbar>3 --> Zelle stirbt)

lustig (interessant bzw. auffällig - nur so nebenbei erwähnt :-) dabei erscheint mir der Widerspruch - das sich lebende Zellen nur mit dem Tod beschäftigen; während sich tote Zellen nur damit beschäftigen wann sie wieder leben dürfen;

die abgekürzte Schreibweise dieser vier Regeln beschäftigt sich allerdings nur mit dem “Leben”:

Wann bleiben lebende Zellen am Leben / Wann werden tote Zellen wieder geboren
23 / 3 (Conway´s Regelwelt)

das ganze in Flash Action Skript 2.0:

```
_global.game_23_3 = function() {
        ////////////////////////////////////
        if (this._visible == 1) {//life
                if (this.moore<2) {
                        this._visible = 0;
                }
                else if (this.moore>3) {
                        this._visible = 0;
                }
        }
        ////////////////////////////////////
        else if (this._visible == 0) {// dead

                if (this.moore == 3) {
                        this._visible = 1;
                }
        }
        _global.borderfunc.call(this);
};
```

ein Problem stellt der endliche Randbereich des ja theoretisch unendlichen Spielfelds dar; es gibt mehrere Möglichkeiten zur Lösung dieses Problems; einerseits ein torusartiges Spielfeld (nach unten “abtauchende” Zellen tauchen oben wieder auf und umgekehrt); oder der Randbereich des Felds wird zur “Todeszone” für die Zellen d.h. alle Zellen im Randbereich sterben (diesen Lösungsansatz verfolgt auch meine cell.life maschine; meine Welt ist eine Scheibe :-)

```
_global.borderfunc = function() {
        ////////////////////////////////////
        //border again
        // left-top
        if (this.arra[0]<=1) {
                this._visible = 0;
        }
        if (this.arra[1]<=1) {
                this._visible = 0;
        }
        // right-bottom
        if (this.arra[0] == 50) {
                this._visible = 0;
        }
        if (this.arra[1] == 50) {
                this._visible = 0;
        }
        ////////////////////////////////////
};
```

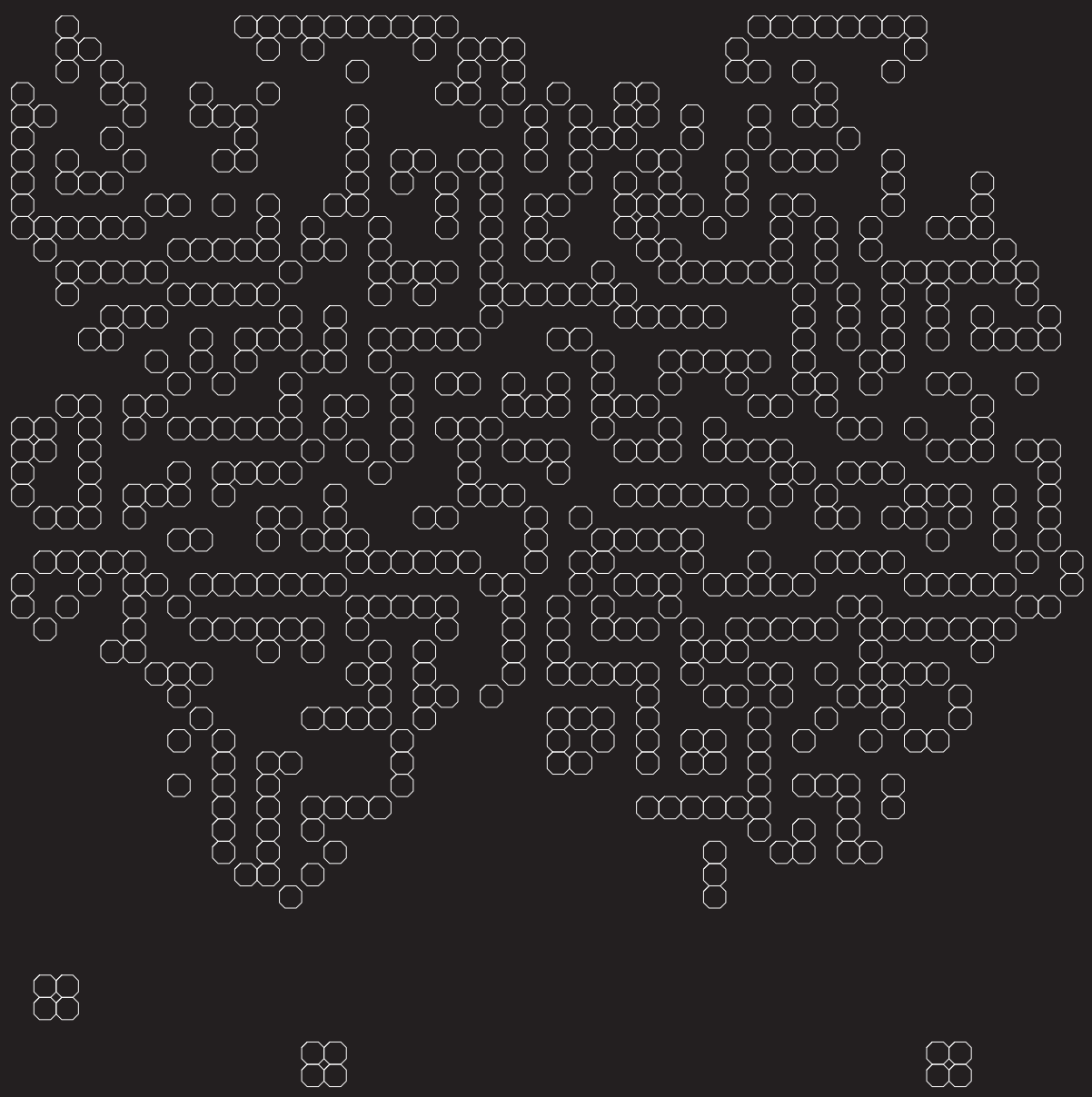
jede Zelle im Randbereich stirbt; allerdings erst nachdem sie die Conway-schen Regeln durchlaufen hat;

der Initiationsproblem (erste “räumliche” Verteilung unbestimmt vieler lebender Elemente/Zellen) wird durch eine einfach Randomfunktion gelöst; d.h. Anfangs leben zufällig viele Zellen in einer zufälligen Konfiguration (dies ist sicher nicht die eleganteste Lösung - aber es war zu seiner Zeit die schnellste Lösung für mich);

```
rnd = random(50);
if (rnd<5) {
        _root[“org_”+x+”_”+y]._visible = 1;
}
```

gesamtes Skript im Anhang;

23/3 system_duration: 24



23/3 system_duration: 68

weiterführend:

Norbert Wiener
(Begründer der Kybernetik)
Stanisław Marcin Ulam
(polnischer Mathematiker; “Erfinder” des zellularen Automaten; Mitarbeiter beim Manhatten Projekt in Los Alamos & und somit Miterfinder der modernen Atombombe; MoteCarlo Methode)
John von Neumann
(Kollege von Ulam; Weiterentwicklung des zelluaren Automaten; 1953 entwickelte er auch die Theorie selbstreproduzierender Automaten (Burcks Hg.; “Theory of self reproducing automata”; 1966))

Abschließend noch das ganze Actionskript meiner cell.life maschine version 1.82 - diese Version basierend auf Conway´s 23/3 Regelwelt - beinhaltet folgende Welten als Dummyfunktionen: 1357/1357 & 34/3; des weiteren ist bitte zu beachten das dieses Skript nur ein erster roher - teilweise sehr umständlich programmierter - Entwurf ist - tja trotzdem - viel Spass :-)

```
//cell.life v.1.82 based on the rules “game of life” by john horton conway
//cell.life v.1.82 programmed by printschler.j.m (action script 2.0)
//
// create the world
_root.onLoad = function() {
	_global.zae = 1;
	_global.tim = 1;
	_root.lif.text = “loading cell.life...”;
	_global.cell_size = 10;
	_global.anz = 51;//World with 50*50 cells
	for (x=1; x<_global.anz; x++) {
		for (y=1; y<_global.anz; y++) {
			_root.createEmptyMovieClip(“org_”+x+”_”+y,_root.getNextHighestDepth());
			_root[“org_”+x+”_”+y]._visible = 0;
			rnd = random(50);
			if (rnd<5) {
				_root[“org_”+x+”_”+y]._visible = 1;
			}
			_root[“org_”+x+”_”+y]._x = (_global.cell_size*x)-_global.cell_size;
			_root[“org_”+x+”_”+y]._y = (_global.cell_size*y)-_global.cell_size;
			_global.zelle.call(_root[“org_”+x+”_”+y]);// function to visualize the cell
			_global.life_func.call(_root[“org_”+x+”_”+y]);// lifefunction for each cell
			_global.borderfunc.call(_root[“org_”+x+”_”+y]);
		}
	}
};
_root.onEnterFrame = function() {
	_global.tim = _global.tim+1;
	_root.lif.text = “23/3 system_duration: “+ _global.tim;


};
////////////////////////////////
// how would the cell look like
_global.zelle = function() {

	this.createEmptyMovieClip(“cell”,1);
	this.arra = new Array();
	this.arra[0] = x;
	this.arra[1] = y;
	with (this.cell) {
		lineStyle(0.25,0xffffff,100);
		moveTo(2.5,0);
		beginFill(0xffffff,0);
		lineTo(7.5,0);
		lineTo(10,2.5);
```

```
                lineTo(10,7.5);
                lineTo(7.5,10);
                lineTo(2.5,10);
                lineTo(0,7.5);
                lineTo(0,2.5);
                endFill();
                lineTo(2.5,0);
        }
        this.cell._yscale = 100;
        this.cell._xscale = 100;
};
////////////////////////////////
_global.life_func = function() {

        this.onEnterFrame = function() {

                // PROGRAMMING THE MATRIX FOR EACH CELL
                if (_root["org_"+(this.arra[0]-1)+"_"+(this.arra[1]-1)]._visible == false) {
                        a = 0;
                }
                if (_root["org_"+(this.arra[0]-1)+"_"+(this.arra[1]-1)]._visible == true) {
                        a = 1;
                }
                if (_root["org_"+(this.arra[0])+"_"+(this.arra[1]-1)]._visible == false) {
                        b = 0;
                }
                if (_root["org_"+(this.arra[0])+"_"+(this.arra[1]-1)]._visible == true) {
                        b = 1;
                }
                if (_root["org_"+(this.arra[0]+1)+"_"+(this.arra[1]-1)]._visible == false) {
                        c = 0;
                }
                if (_root["org_"+(this.arra[0]+1)+"_"+(this.arra[1]-1)]._visible == true) {
                        c = 1;
                }
                ////////////////////////////////////////////////////////////
                if (_root["org_"+(this.arra[0]-1)+"_"+(this.arra[1])]._visible == false) {
                        d = 0;
                }
                if (_root["org_"+(this.arra[0]-1)+"_"+(this.arra[1])]._visible == true) {
                        d = 1;
                }
                if (_root["org_"+(this.arra[0]+1)+"_"+(this.arra[1])]._visible == false) {
                        e = 0;
                }
                if (_root["org_"+(this.arra[0]+1)+"_"+(this.arra[1])]._visible == true) {
                        e = 1;
                }
                ////////////////////////////////////////////////////////////
                if (_root["org_"+(this.arra[0]-1)+"_"+(this.arra[1]+1)]._visible == false) {
                        f = 0;
```

```
                }
                if (_root["org_"+(this.arra[0]-1)+"_"+(this.arra[1]+1)]._visible == true) {
                        f = 1;
                }
                if (_root["org_"+(this.arra[0])+"_"+(this.arra[1]+1)]._visible == false) {
                        g = 0;
                }
                if (_root["org_"+(this.arra[0])+"_"+(this.arra[1]+1)]._visible == true) {
                        g = 1;
                }
                if (_root["org_"+(this.arra[0]+1)+"_"+(this.arra[1]+1)]._visible == false) {
                        h = 0;
                }
                if (_root["org_"+(this.arra[0]+1)+"_"+(this.arra[1]+1)]._visible == true) {
                        h = 1;
                }
                ////////////////////////////////////////////////////////
                this.moore = (a+b+c+d+e+f+g+h);
                trace(this.moore);
                ////////////////////////////////////////////////////////
                // call the life function
                _global.game_23_3.call(this);// based on conways original
                //_global.game_1357_1357.call(this);
                //_global.game_34_3.call(this);


        };
};
_global.borderfunc = function() {
        ////////////////////////////////////
        //border again
        // left-top
        if (this.arra[0]<=1) {
                this._visible = 0;
        }
        if (this.arra[1]<=1) {
                this._visible = 0;
        }
        // right-bottom
        if (this.arra[0] == 50) {
                this._visible = 0;
        }
        if (this.arra[1] == 50) {
                this._visible = 0;
        }
        ////////////////////////////////////
};
```

```
//LIFE FUNCTION
////////////////////////////////////
// conways rules of life a 23/3 system
_global.game_23_3 = function() {
        ////////////////////////////////////
        if (this._visible == 1) {//life
                if (this.moore<2) {
                        this._visible = 0;
                }
                else if (this.moore>3) {
                        this._visible = 0;
                }
        }
        ////////////////////////////////////
        else if (this._visible == 0) {// dead

                if (this.moore == 3) {
                        this._visible = 1;
                }
        }
        _global.borderfunc.call(this);
};
////////////////////////////////////
```

Quellenverzeichnis Internet:

http://www.metatektur.org (stand: 141110)
http://www.stubenmusicstudio.com/metatecture/?p=82 (stand: 091110)
http://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens (stand: 13.11.10)
http://de.wikipedia.org/wiki/John_Horton_Conway (stand: 13.11.10)
http://de.wikipedia.org/wiki/Stanis%C5%82aw_Marcin_Ulam(stand: 13.11.10)

Quellenverzeichnis Printmedien:

Norbert, Elias/Michael Schröter(Hg.): Arbeiten zur Wissenssoziologie - 2. Über Die Zeit, Frankfurt am Main 1989