

Generování bludiště, buněčné automaty

Ivana Kolingerová

Obsah:

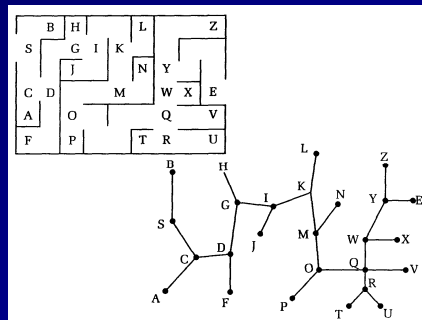
1. Generování bludiště
2. Buněčné automaty

Literatura:

Francis S.Hill Jr.: Computer Graphics,
Macmillan Publishing Company, New
York, 1990

1. Generování bludiště

- pravoúhlé bludiště $R \times C$ buněk, náhodné, ale cesta musí existovat
- popis např. grafem: uzel odpovídá buňce, v listu cesta končí, z vnitřního uzlu pokračuje, stupeň uzlu odpovídá větvení
- cyklu v bludišti odpovídá cyklus v grafu (obvykle nepoužijeme)



- reprezentace v programu: např. pamatovat pro každou buňku, zda je její severní a východní stěna plná

```
var north_wall, east_wall: array[0..num_rows,0..num_cols]
    of boolean;
```

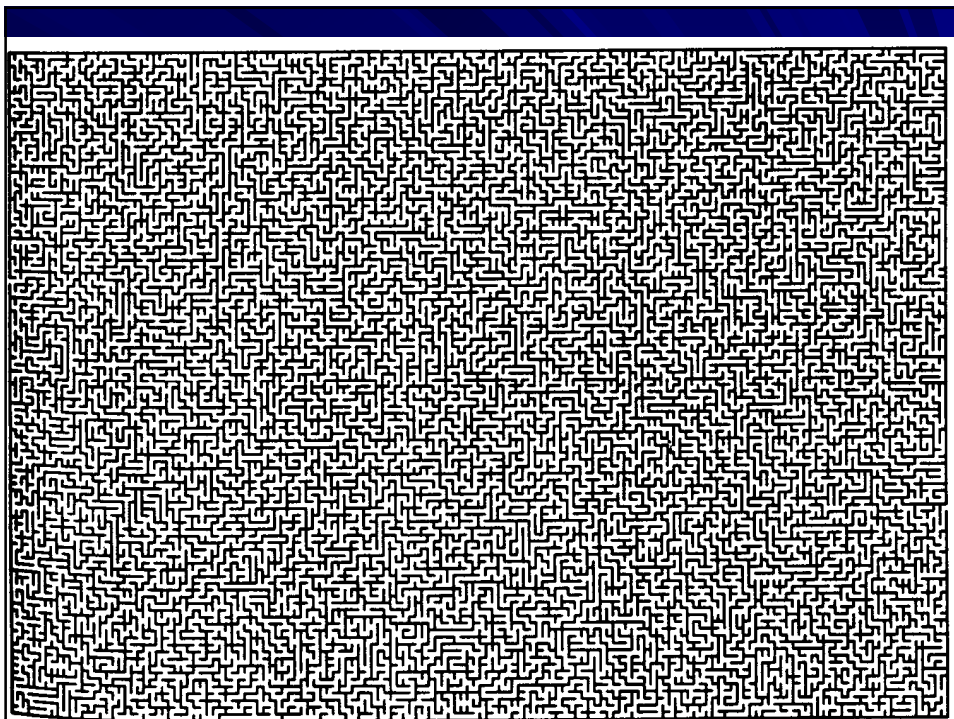
=> `north_wall[i,j]=true` <=> buňka i,j má severní stranu
"bez dveří"

Řádek `num_rows`: fantomová řádka buněk pod bludištěm,
obsahující informaci o dolní hraně bludiště

Dtto sloupec `num_cols` – pravá hranice

Generování bludiště

- Začít se všemi stěnami plnými, takže bludiště nemá žádné průchody.
- Přemístit se do náhodné buňky, úkolem je probourat se do sousedních buněk:
 - zkontrolovat 4 sousední buňky, zda mají všechny stěny nedotčené. Pokud ne, buňka už byla navštívena a je zařazena v nějaké cestě.
 - Z nenavštívených buněk se jedna vybere náhodně a spojovací zeď do ní se zbourá. Ostatní kandidátky se uloží do zásobníku.
- Po přechodu do nově připojené buňky se pokračuje dál stejně.
- Když neexistuje sousední nenavštívená buňka, vybere se pokračovací buňka ze zásobníku.
- Když prázdný zásobník, všechny buňky navštíveny.



KPG1-7

Na konec výběr počáteční a koncové buňky S, T
(obvykle na hraně bludiště)

Upozornění: novou buňku je vhodné propojit
s již navštívenou sousedkou, jinak vzájemně izolované cesty.

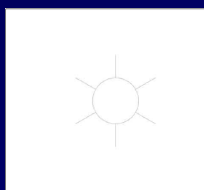
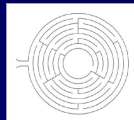
Varianty – propojit vždy x někdy (pak pozor na existenci cesty
mezi S a T)
- občas "vybourat" něco navíc (tak 1-20 x)

Průchod bludiště

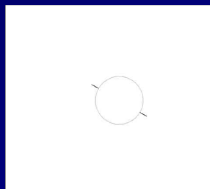
- backtracking – pohyb v náhodném směru s návratem
a označením chybné cesty ("zazdít")
- místo náhodného směru možno např.
"stále vlevo", pak při S a T uvnitř nemusí
zvládnout smyčky

KPG1-8

Kruhové bludiště

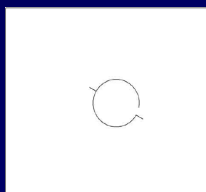


1. Vyberte max. počet radiálních
stěn N na 1 vrstvu, $N \geq 2$ (zde
 $N=6$)

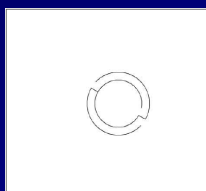


2. Vyberte náhodně, které z N
radiálních stěn budou
v 1.vrstvě (alesp.1) a nakreslete
je

KPG1-9

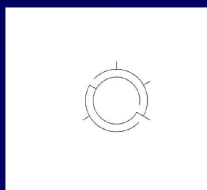


3. Vyberte náhodně jednu z radiál. stěn, namalujte od jejího vnitřního okraje kružnici ve směru CW, skončete o kousek dřív

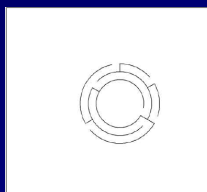


4. Pro všechny radiál. stěny: namalujte oblouk CCW směrem k následující stěně, skončete kousek před ní

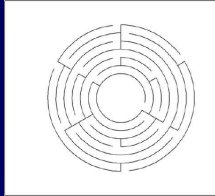
KPG1-10



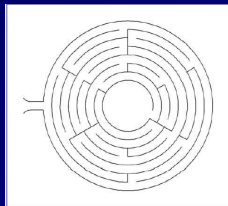
5. Vyberte náhodně, které z N radiálních stěn bude přítomno v příští vrstvě (alesp.1), a namalujte je



6. Nakreslete oblouky od vnějších konců radiál. stěn ve směru CW



7. Opakujte proces přidávání radiál.stěn a oblouků, střídejte CW a CCW oblouky, až velikost bludiště odpovídá vaší představě

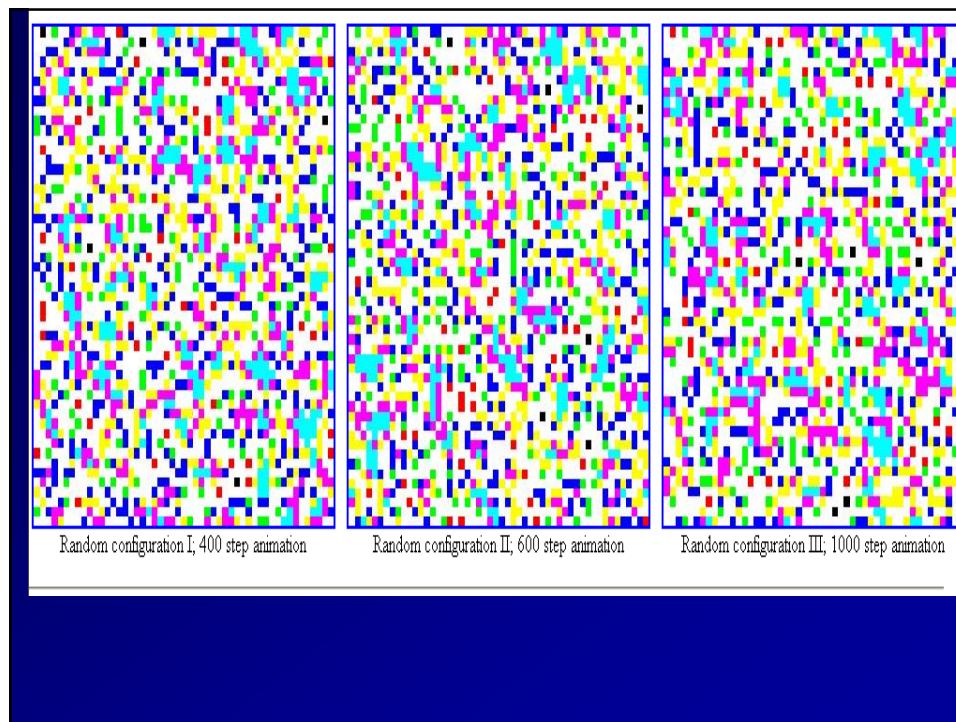


8. Přidejte jednu kruhovou vrstvu navíc s otvorem na libovol. místě

2. Buněčné automaty

- Simulace reálného života
- John Conway (1970)
- Game of Life – 2D binární buněčný automat
- každá generace – nekonečná šachovnice buněk, kde každá buňka je živá nebo mrtvá
- stav buňky v čase t \rightarrow generace $t+1$ podle stavu buňky a 8-okolí

Př.: <http://world.std.com/~bgw/applets/1.02/Life/Life.html>

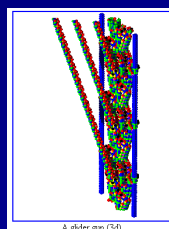


KPG1-14

Obvyklá pravidla:

- Buňka živá v čase t bude živá v čase $t+1 \iff$ má v čase t dva nebo tři živé sousedy
- Buňka mrtvá v čase t bude živá v čase $t+1 \iff$ má v čase t tři živé sousedy
- Ostatní buňky umírají nebo zůstávají mrtvé

Náhled – pomocí animací nebo skládání následujících generací na sebe do 3D obrázku



KPG1-15

Buňky lze také obarvovat podle počtu sousedů nebo stáří



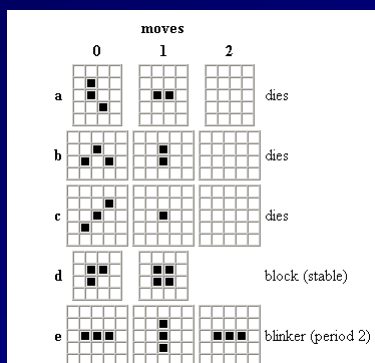
KPG1-16

Pozor, všechna narození a úmrtí nutno řešit simultánně
– nutno oddělit vstupní a výstupní data !

Pro implementaci zvětšit matice o 1 na každou stranu

Ukázky

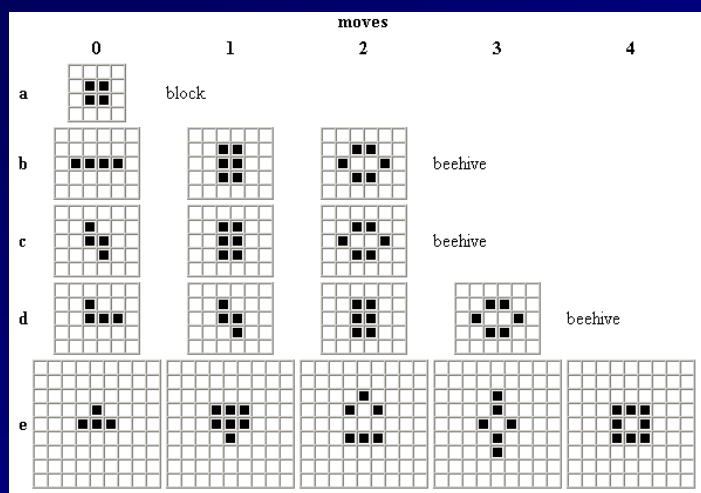
Základní konfigurace:



KPG1-17

Ukázka: program Blinkers aj.

Základní konfigurace:



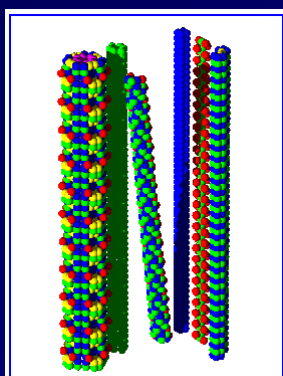
KPG1-18

Základní konfigurace:

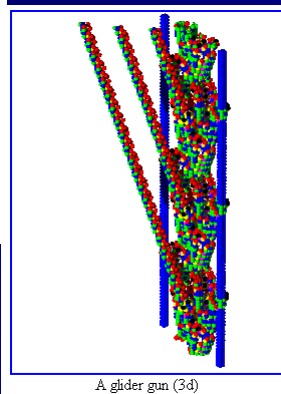
Table 1. The long-term behavior of the Game of Life when the initial configuration is a filled square with sides of length n .

$n=1$	dies immediately
$n=2$	is stable from $t=0$ (it is a block)
$n=3$	dies at $t=9$
$n=4$	dies at $t=4$
$n=5$	terminates in the honey farm at $t=11$
$n=6$	forms a pond at $t=5$
$n=7$	makes a traffic light at $t=11$
$n=8$	dies at $t=6$
$n=9$	turns into 4 traffic lights and 1 tub at $t=17$
$n=10$	forms 1 pond at $t=17$
$n=11$	becomes a traffic light at $t=33$
$n=12$	fades at $t=9$
$n=13$	makes a tub at $t=18$
$n=14$	turns into a pond at $t=9$
$n=15$	dies at $t=22$
$n=16$	forms 4 ponds and 4 beehives at $t=11$
$n=17$	makes 1 tub, 8 blocks, and 12 blinkers at $t=33$
$n=18$	turns into 8 blocks, and 1 pond at $t=17$
$n=19$	makes 5 traffic lights at $t=21$
$n=20$	dies at $t=12$

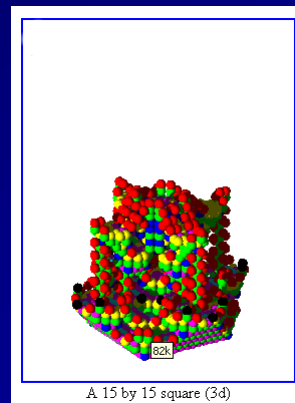
Některé 3D obrázky:



Some periodic patterns and a glider (3d)

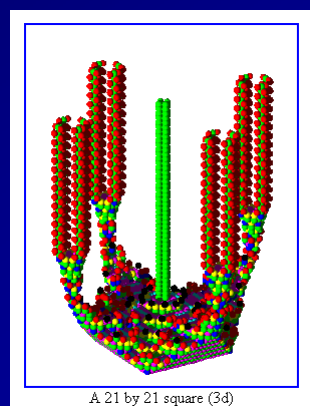
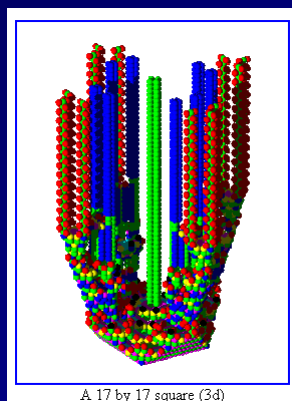
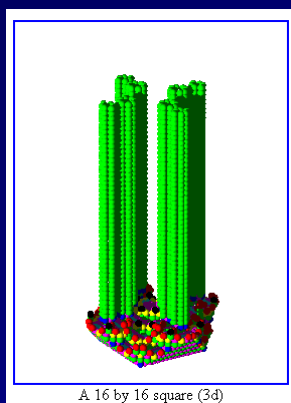


A glider gun (3d)

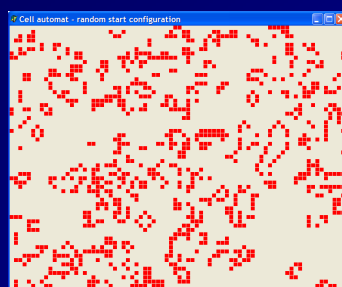


A 15 by 15 square (3d)

Některé 3D obrázky:



Zajímavé je také začít s náhodnou počáteční konfigurací



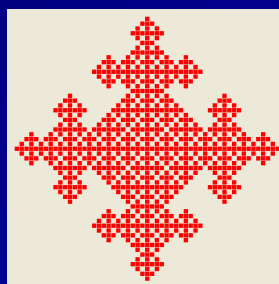
Ukázka: program Randconfig

Modifikace:

- chování založené na jednom defektu a modifikaci pravidel

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Typ 1: 1 vznikne, pokud čtverec
ortogonálně sousedí s 1x1,
žádná smrt

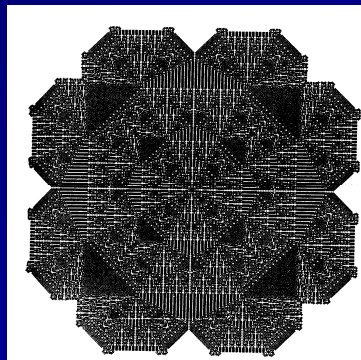


Ukázka: program 1defect

Typ 2 (mod 2): 1 vznikne, pokud čtverec
ortogonálně sousedí s 1x1 pro sudé t
nebo s 1x1 ortogonálně i diagonálně pro liché t,
žádná smrt

(mod 6): totéž s mod 6

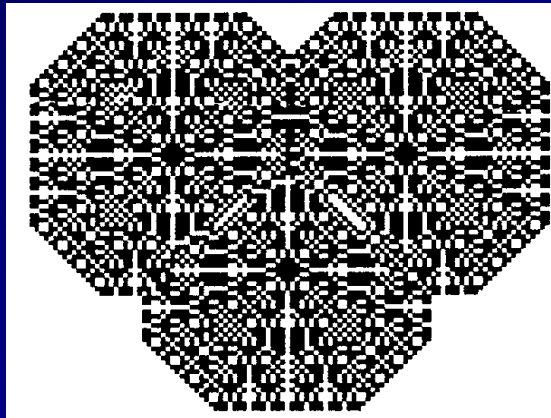
Typ 2



Typ 3: více defektů, rostou, pak soutěží (+úmrtí buněk)

Typ 4. větší lokální okolí

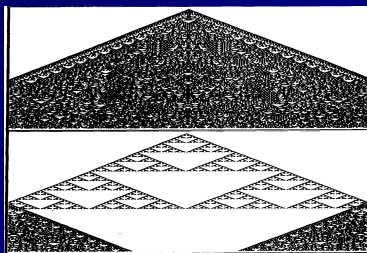
.... atd.



Typ 3 (3 defekty)

Jednorozměrné buněčné automaty

- buňka a 2 sousedi na každou stranu
- zápis pravidla: $(d_0, d_1, d_2, d_3, d_4; a_0, a_1, a_2, a_3, a_4)$
- 0 v d_i – pokud je střední buňka mrtvá
 - a i buněk ze 4 sousedů živé, centrální buňka zemře
- 1 v d_i - pokud je střední buňka mrtvá
 - a i buněk ze 4 sousedů neživých, centrální buňka ožije
- a_i – totéž pro živý střed

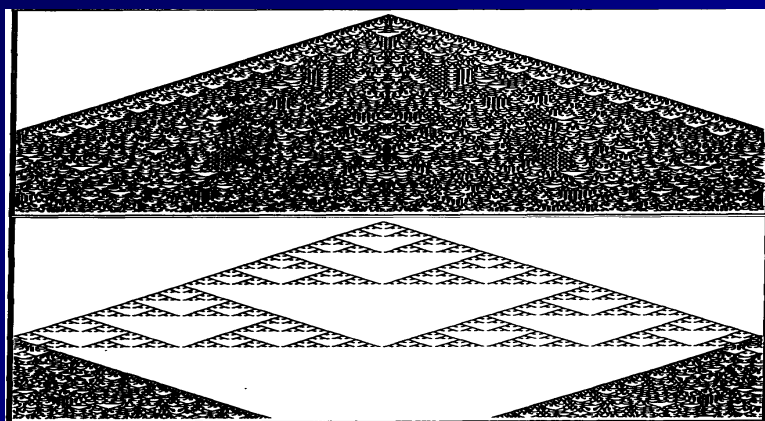


Př.: $(0, 1, 1, 0, 0; 0, 0, 1, 0, 0)$ - živá buňka v příš. generaci, pokud stř. buňka mrtvá a právě (1-2) sousedi neživí nebo stř. buňka živá a právě 2 sousedi neživí

KPG1-27

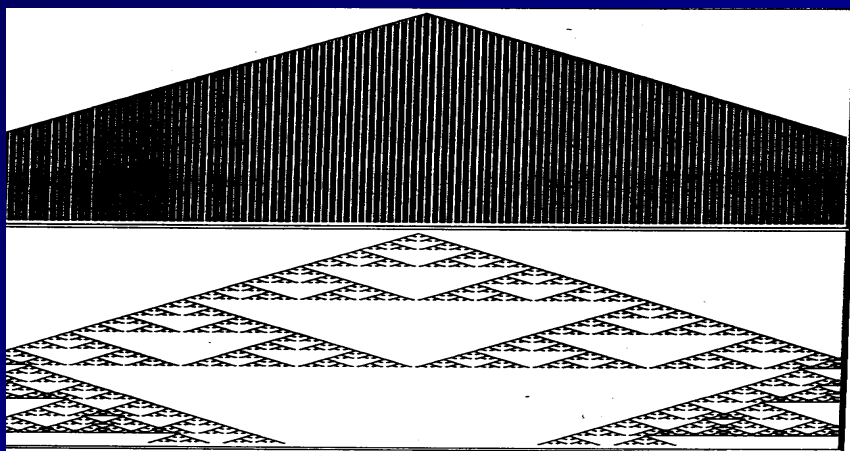
Př.: (0,1,1,0,0;0,0,1,0,0)

ze 2 sousedních živých buněk vznikne Sierp.trojúhelník,
z 1 buňky "vzorkovaný trojúhelník"
řešeno jako "wrap-around"

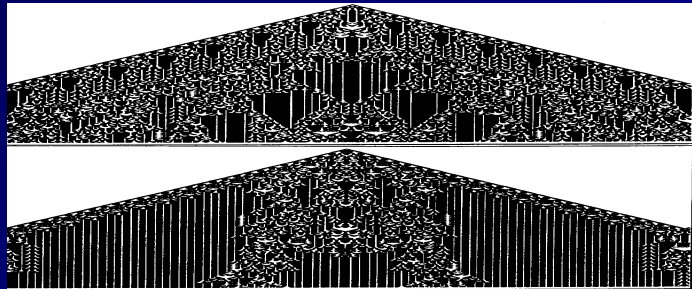


KPG1-28

Př.: (0,1,1,0,0;0,0,1,1,0) – z 1 buňky proužky,
ze 2 Seirpin.trojúhelník



Př.: (0,1,1,0,0;0,1,0,1,1) – ze 3 sousedních buněk – objeví se "lebka"



Literatura:

Martin Gardner: Mathematical Games. The Fantastic Combinations of J. Conway's New Solitaire Game "Life", Scientific American 223 (October 1970), pp.120-123

John.E.Pulsifer, Clifford A. Reiter: One Tub, Eight Blocks, Twelve Blinkers and Other Views of Life, Computers and Graphics, Vol. 20, No. 3, pp.457-462, 1996, Elsevier Science

KPG1-31

Elwyn R. Berlekamp, John H. Conway,
Richard K. Guy: Winning Ways for
Your Mathematical Plays, Academic
Press, 1982

Michael Frame: Sensitivity in Cellular
Automata: Some Examples,
Computers and Graphics, Vol.18, No.
5, pp.733-737, Elsevier Science

C.A.Pickover: Mathematics and Beauty –
VIII, Tessellation Automata Derived
from a Single Defect, Computer Math
Applic., Vol. 17, No. 1-3, pp.321-326,
1989, Pergamon Press