JAPLA 研究会資料 2011/7/23

ライフゲームーその１
ーJの素朴なプログラムー

西川　利男

JAPLA の例会で２回にわたって、志村正人氏より「タートル・グラフィックス」、「プログラムされた虫」のパズルとそのＪプログラムについての発表があった。

掲載された「数学ゲーム」[1]の本には、コンウェイの有名な「ライフゲーム」も載っている。

[1] マーチン・ガードナー著、一松信訳「マーチン・ガードナーの数学ゲームＩ」
別冊日経サイエンス、日経サイエンス社(2010).

ライフゲームは以前にもちょっと手がけたことがあるが、今回あらためて２回にわけてゆっくりやってみる。

第１回目は、まずは素朴な方法でＪのプログラムを作った。それを元として、第２回目には、Ｊの grid を利用した OOP プログラミングの例として、蓼科での実験テーマに向けて、作成、準備している。ご期待のほどを！

１．いろいろな平面図形パズル

志村氏の「タートル・グラフィックス(ロゴ)」と「プログラムされた虫」、そして「ライフゲーム」など２次元の平面図形を使うパズルにはいろいろなものがある。

空飛ぶ鳥の目から見れば（３次元鳥瞰図、bird view）、全体のようすもすぐ分かるものの、地上を這うアリの目（２次元平面）からはまったく分からないというものである。このような考え方で、次のように分類してみた。

盲（めくら）のゲーム……タートル・グラフィックス
目明きのゲーム（目で見て判断）……ライフゲーム、プログラムされた虫

ローカル座標、グローバル座標の切り替えの問題も含めて、「タートル・グラフィックス」でどういう図形が出来上がるかを予想することは難しい。また、「プログラムされた虫」も同様な意味で、志村氏は極めて難しいプログラミングを強いられたことになったのだろう。

一方、「ライフゲーム」も似てはいるが、周囲の環境を目で見て、規則にしたがって、次の動きを決めるゲームであることから、プログラム処理はずっと容易である。

２．ライフゲームとその規則

純粋数学者のコンウェイ(John Horton Conway)が発明した「ライフゲーム」は生物の誕生や死滅を模した、一種のシミュレーションゲームである。

方眼のチェッカー盤に適当な数の石を置く。次にそれぞれのチェッカー盤のマス目について以下の規則にしたがって、石を増やしたり、減らしたりする。

生存　　周囲に２個、３個の石があれば、そのマス目の状態はそのまま存続する
死滅　　周囲に０個、１個の石があれば、その石は死滅する
　　周囲が４個以上でも、その石は死滅する
誕生　　周囲に３個の石があれば、そのマス目で石が誕生する

これを、チェッカー盤のすべてのマス目について調べ、処理を行って１つの状態とする。次にこれを元にさらに上の処理を行って次の状態とする。これを次々と続ける。

このようにして続けていくと、途中にいろいろ興味深いパターンが現れる。これがライフゲームである。

３．ライフゲームのＪプログラム

３．１　準備と簡単な例

ライフゲームのデータは１と０とから成る２次元の配列で示す。

まず、ある対象とする１つのセルの周囲を参照するインデックスが必要である。

```
indx =: 3 : 0
Y =. <"(0) y.
Y (,L:0) /"(0 1) Y
)
NB.    indx <: i. 3
NB. +-----+----+----+
NB. |_1 _1|_1 0|_1 1|
NB. +-----+----+----+
NB. |0 _1 |0 0 |0 1 |
NB. +-----+----+----+
NB. |1 _1 |1 0 |1 1 |
NB. +-----+----+----+
```

これを用いて、あるセルの周囲の値を取り出す関数 env を定義する。左引数のテスト配列に対して、右引数で指定したセルの周囲の値を取り出す。

```
env =: 3 : 0
:
BX =: indx <: i.3
ind =. (,(<y.) +L:0 BX) -. (<y.)
ind { x.
)
NB.     i. 4 4                テストする配列
NB.  0  1  2  3
NB.  4  5  6  7
NB.  8  9 10 11
NB. 12 13 14 15
NB.     (i. 4 4) env (1, 1)    配列の要素(1, 1)の周囲の値を返す
NB. 0 1 2 4 6 8 9 10
NB.     (i. 4 4) env (1, 2)    配列の要素(1, 2)の周囲の値を返す
NB. 1 2 3 5 7 9 10 11
```

セルの１つの要素に対する先のライフゲームの規則は関数 lif で与えられる。

```
lif =: 3 : 0
:
ST =. +/ x. env y.                NB. セルの周囲の値の和
select. ST
  case. 0, 1 do. 0                NB. 死滅
  case. 2    do. (<y.) { x.       NB. 生存
```

```
  case. 3     do. 1              NB. 誕生
  fcase.      do. 0              NB. 死滅
end.
)
```

これをすべてのセルの要素にわたって、行う関数が life である。

```
life =: 3 : 0
DA =. y.
'M N' =. $ DA
DB =. }.^:(1) }. (<:N) {.^:(1) (<:M) {. M index N
DC =. > DA lif L:0 DB
(0,^:(1) 0, DC, 0) ,^:(1) 0
)
```

簡単な配列 DA0 でやってみる。

```
   DA0
0 0 0 0 0
0 1 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 0 0
```

これに関数 life を操作してみる。

```
   life DA0
0 0 0 0 0
0 0 0 0 0
0 1 1 0 0
0 0 0 0 0
0 0 0 0 0
```

もう１回、関数 life を操作してみる。

```
   life^:(2) DA0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

すなわち、最初の曲がったデータの配列は life 操作で、ヨコの直線になり、さらに life 操作すると、死滅してしまった。

３．２　便利なツールと実際のライフゲーム

今の場合は、ライフゲームの初期値は１と０の配列で与えるが、次のイディオム的定義が便利である。

```
DAX =: ] ;._2 (0 : 0)
0 0 0 0 0
0 0 1 0 0
0 1 1 1 0
```

```
0 0 0 0 0
)
```

ただし、これは数字で表した文字列になるので、(".)により数値に直すことが必要である。

```
   $DAX
4 10
   $".DAX
4 5
```

ライフゲームの初期状態となる値の配列はそれほど大きくなくても、ライフゲーム操作を続ける従い、パターンが移動してゆくことがよく起きる。そのため相当大きな配列が必要になる。このために、初期状態の小さなパターン配列を中に埋め込んで拡大する関数 expand を作った。

```
expand =: 3 : 0
:
'm n a b' =. x.
MA =. y.
MB =. m {. 0 ,^:(a) MA
MC =. 0 ,"(1)^:(b) MB
n {."(1) MC
)
```

いまの例では、(11x16)の配列の中に、(3 5)の位置を左上にして、入れて拡大した。

```
   (11 16 3 5) expand (". DAX)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 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 0 0 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

さらに、0と1との代わりに文字'_'と'*'とを使って、これを見やすくした。

```
   ((11 16 3 5) expand (". DAX)) {'_*'
________________
________________
________________
________________
_______*________
______***_______
________________
________________
```

```
________________
________________
________________
```

これを初期パターンとして、ライフゲーム操作を6回、行った結果である。

```
(life^:(6) (11 16 3 5) expand (". DAX)) {'_*'
```

```
________________
________________
________________
_______*________
______***_______
_____**_**______
______***_______
_______*________
________________
________________
________________
```

さらに続けてみる。

```
(life^:(7) (11 16 3 5) expand (". DAX)) {'_*'
```

```
________________
________________
________________
______***_______
_____*___*______
_____*___*______
_____*___*______
______***_______
________________
________________
________________
```

```
(life^:(8) (11 16 3 5) expand (". DAX)) {'_*'
```

```
________________
________________
_______*________
______***_______
_____*_*_*______
____***_***_____
_____*_*_*______
______***_______
_______*________
________________
________________
```

このように、パターンはいろいろと変化していく。これを生物システムの栄枯盛衰として眺めると興味深いものである。

```
NB. Life Game
NB.  programmed by Toshio Nishikawa
NB.  2011/6/1

indx =: 3 : 0
Y =. <"(0) y.
Y (,L:0) /"(0 1) Y
)
NB.   indx <: i. 3
NB. +-----+----+----+
NB. |_1 _1|_1 0|_1 1|
NB. +-----+----+----+
NB. |0 _1 |0 0 |0 1 |
NB. +-----+----+----+
NB. |1 _1 |1 0 |1 1 |
NB. +-----+----+----+


env =: 3 : 0
:
BX =: indx <: i.3
ind =. (,(<y.) +L:0 BX) -. (<y.)
ind { x.
)

NB.    i. 4 4
NB.  0  1  2  3
NB.  4  5  6  7
NB.  8  9 10 11
NB. 12 13 14 15
NB. Usage:
NB.    (i. 4 4) env (1, 1)
NB. 0 1 2 4 6 8 9 10
NB.    (i. 4 4) env (1, 2)
NB. 1 2 3 5 7 9 10 11

lif =: 3 : 0
:
ST =. +/ x. env y.
select. ST
  case. 0, 1 do. 0
  case. 2    do. (<y.) { x.
  case. 3    do. 1
  fcase.     do. 0
end.
```

```
)
life =: 3 : 0
DA =. y.
'M N' =. $ DA
DB =. }.~(1) }. (<:N) {.~(1) (<:M) {. M index N
DC =. > DA lif L:0 DB
(0,~(1) 0, DC, 0) ,~(1) 0
)


NB. number matrix data
DA0 =: 5 5$0 0 0 0 0, 0 1 0 0 0, 0 1 0 0 0, 0 0 1 0 0, 0 0 0 0 0

NB. DA1 is character matrix (5 x 9)
NB. if you want number matrix, then convert as ~. DA1
DA1   =: ] ;._2 (0 : 0)
0 0 0 0 0
0 1 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 0 0
)

NB. Life Game Data /ライフゲーム  p.34-5, Fig. a, b, d, e ========
LFAa =: ] ;._2 (0 : 0)
0 0 0 0 0
0 1 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 0 0
)

LFBb =: ] ;._2 (0 : 0)
0 0 0 0 0 0
0 0 0 0 0 0
0 1 1 1 1 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
)

LFBe =: ] ;._2 (0 : 0)
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 1 0 0 0 0
0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
)

LF_glider =: ] ;._2 (0 : 0)
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0
0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
)



NB. Examples
NB.    (life^:(5) ". LFBe) {'_*'
NB.    (life^:(7) ". LFBe) {'_*'

expand =: 3 : 0
:
'm n a b' =. x.
MA =. y.
MB =. m {. 0 ,^:(a) MA
MC =. 0 ,"(1)^:(b) MB
n {."(1) MC
)

NB. Uasage:
NB.   (10 16 3 4) expand (". DA1)
NB. 8 8 8 8 9 9 9 9 9 0 0 0 0 0 0 0
NB. 8 8 8 8 9 9 9 9 9 0 0 0 0 0 0 0
NB. 8 8 8 8 9 9 9 9 9 0 0 0 0 0 0 0
NB. 8 8 8 8 0 0 0 0 0 0 0 0 0 0 0 0
NB. 8 8 8 8 0 1 0 0 0 0 0 0 0 0 0 0
NB. 8 8 8 8 0 1 0 0 0 0 0 0 0 0 0 0
NB. 8 8 8 8 0 0 1 0 0 0 0 0 0 0 0 0
NB. 8 8 8 8 0 0 0 0 0 0 0 0 0 0 0 0
```

```
NB. 8 8 8 8 0 0 0 0 0 0 0 0 0 0 0 0
NB. 8 8 8 8 0 0 0 0 0 0 0 0 0 0 0 0

NB.   (life^:(0) (11 16 1 2) expand (". LFBe))
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

NB.   (life^:(8) (11 16 1 2) expand (". LFBe))
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
NB. 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0
NB. 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0
NB. 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0
NB. 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NB. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

NB.   (life^:(8) (11 16 1 2) expand (". LFBe)) {'_*'
NB. ________________
NB. ________________
NB. ______*_________
NB. _____***________
NB. ____*_*_*_______
NB. ___***_***______
NB. ____*_*_*_______
NB. _____***________
NB. ______*_________
NB. ________________
NB. ________________
```