

# タートルのサーキット走行 リアルタイムシュミレーションの描画 (J6 版)

Masato SHIMURA  
jcd02773@nifty.ne.jp

2007 年 10 月 29 日

## 目次

1	同じサイズのタートル	1
1.1	サーキットの作成 . . . . .	1
1.2	ASEP と ZRP . . . . .	3
1.3	ローテーション . . . . .	4
2	異なるサイズのタートル	5

### 概要

タートルグラフィックスで ASEP と ZRP を用いて、タートルをサーキットでリアルタイムに走行させて、シュミレーションを行う。

## 1 同じサイズのタートル

### 1.1 サーキットの作成

ここでは同一のサイズのタートルとする。  
30×4×2 のグリッドを作成する。

タートルの形 正 6 角形が見やすい

```
J0=. repeat 6 fd 1 rt 60 NB. Hexagon
```

先頭と後尾 左上の縦を先頭グリッドとする。時計回り

計算では右端が先頭、左端が最後尾となっている。グラフィックスでは左が前方としている。最初に計算データをリバースする。

```
TMP0=. anal_turtle_circuit y
```

```
TMP=:|. "1 ; COUNTER{ TMP0 NB. last to top
```

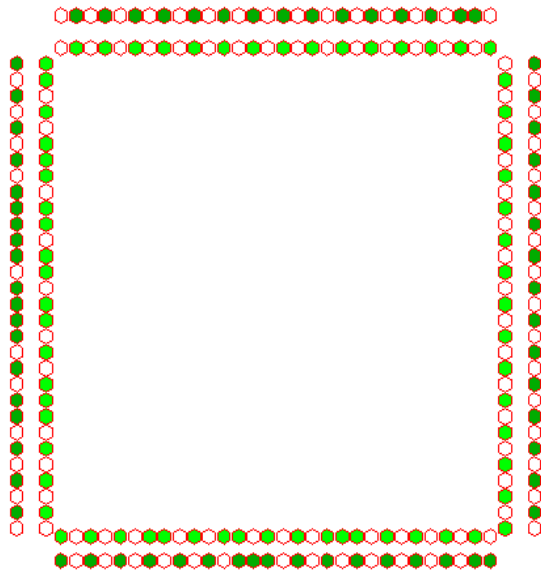


図1 circuit

横  $30 \times 2$  タートルグラフィックスで星をちりばめる時の位置の指定は複素数で行う。

横に4列(上2、下2)

```
4{.( "1) a=. 0 4 66 70 +/ j.3+2*>:i.30
```

```
0j5 0j7 0j9 0j11      0j57 0j59 0j61 0j63
4j5 4j7 4j9 4j11      4j57 4j59 4j61 4j63
```

```
66j5 66j7 66j9 66j11  66j57 66j59 66j61 66j63
70j5 70j7 70j9 70j11  70j57 70j59 70j61 70j63
```

縦  $30 \times 2$  縦の位置の指定

j.>: 3 6 64 68 NB. 列の位置

0j4 0j7 0j65 0j69 NB. 複素数で

```
2 15 $ 30{.6+2*i.60 NB.30 台を一つづつ離して
6 8 10 12 14 16 18 20 22 24 26 28 30 32 34
36 38 40 42 44 46 48 50 52 54 56 58 60 62 64
```

(30{.6+2\*i.60) +/ j. >: 0 3 64 68 NB. 縦に組み上げる

```
6j1 6j4 6j65 6j69
```

```

8j1 8j4 8j65 8j69
10j1 10j4 10j65 10j69
. . . .
. . . .
58j1 58j4 58j65 58j69
60j1 60j4 60j65 60j69
62j1 62j4 62j65 62j69
64j1 64j4 64j65 64j69

```

走行方向 時計回りとする。方向は図示する。(上下、左右は逆になる。)  
 走行、追い越しは、サーキットではインサイドが走行になる。

```
show ;("1),.(B0 start {CLR2},A0 start {CLR3 NB. reverce lane
```

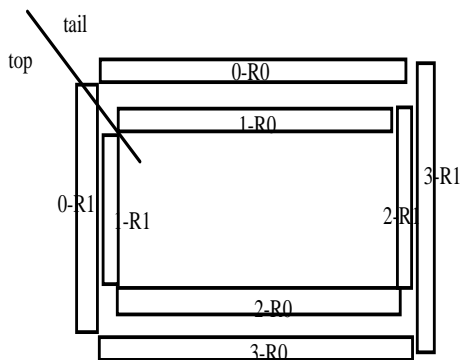


図 2 circuit

```

R0=.0 4 66 70 +/ j.3+ 2* >:i.30 NB. tate
R1=. (30{. 6+ 2* i. 60)+/ j.>: 0 3 64 68 NB. yoko
A0=. ({."1 R1),({: R0),(|. {:"1 R1),|. {. R0 NB. ourtt lane (oikoshi)
B0=. (1{"1 R1),(2{ R0),(|.2{"1 R1),|.1{ R0 NB. inner lane (soukou)

```

## 1.2 ASEP と ZRP

仮想サーキット 自走型の ASEP を用いて、メモリの中に仮想サーキットを作成する。ZRP は慎重な型とする。

先頭の前方 最後尾のグリッドを馬の鼻面の人参のように前に置きコントロールとする。  
グリッド タートルがグリッドをはみ出さないよう(脱落になる)に留意する  
レーン変更 ZRP 実施時に色を変更する。

### 1.3 ローテーション

ローテーションのアルゴリズム

```
forward_sub2_circuit=: 3 : 0
NB. self run engine for circuit
NB. forward_sub1 is same as straight
TX =. forward_asep_sub1_circuit y
MKR=. ;({: L:0 y),{. y NB. 0 0,1 0.0 1,1,1-->2 #. is 0/2/1/3
NB. marker
select. 2 #. MKR NB. 2 notation
  case. 2 do. NB. 1 0 is 2 --> may go
    TX=. ({: TX),}.:TX NB. remove last to top
    fcase. do. TX=. }.: TX
  end.
NB. not forward case// recursive
NB. pecurier circuit /over zone--> assume 1
)
```

```
forward_asep_sub1_circuit =: 3 : 0
NB. always same size
NB. calc ASEP
NB. self forward type
NB. x is 0/1 by probability
IND=. 2 = 2(+/\) \ y,{. y
COMPARE=. (bare ({@> 1 2) = L:0 +/\ y, :IND),.0
+/\ ({: COMPARE),: _1|. {. COMPARE
)
```

チェック箇所	( { : L:0 y),x	先頭, 最後尾
分岐	1 0 先頭は前進 TX=. ( { : TX), }. } :TX 先頭を最後尾の空きグリッドに移し 替える。次回の先頭と最後尾は交代 する。 他 先頭は前進しない (TX=. } :TX)	2進法で照査 1 0 -> 2

先頭は最後尾により押さえられている

```

a, : 1 forward_sub2_circuit a
0 0 1 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1    NB.-->forward
0 0 1 0 1 0 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 1

```

先頭の邪魔は無いので、先頭は最後尾のグリッドに進む。

```

a, : 0 forward_sub2_circuit a
0 0 1 1 0 1 0 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1    NB.-->forward
1 0 1 0 1 0 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 0

```

## 2 異なるサイズのタートル