J 研究会資料　2006/3/25

J のオブジェクト指向プログラミングーその４

―J-Grid による数独パズルをもっと使いやすく―

西川　利男

J の Grid は Excel に相当するスプレッド・シートだが、J ユーザにとってはその内部が分かるだけに、ずっと便利な環境である。また最新技術のオブジェクト指向プログラミングとはどんなものかを身近に体験できるメリットもある。

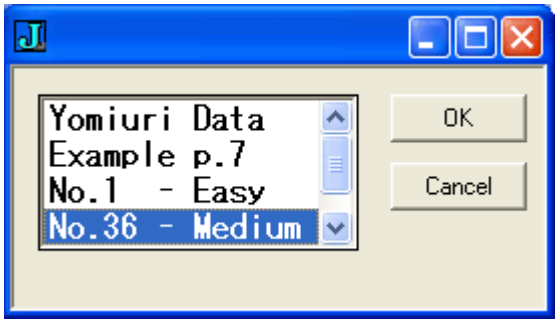
数独パズルが今、話題を呼んでいるが、昨年暮れ、APL/J シンポジウムで J の Grid プログラムの例として紹介した[1], [2]。さらに、R.Hui により解を求める J のプログラムが出され、この解説を行った[3]。その後、筆者自身、J―Grid プログラミングへの理解もさらに深まり、今回もっと使い勝手を良くしたので報告する。

現在のプログラミング言語の傾向として、従来の数値計算、あるいは検索、ソートなどの事務処理を行うだけでなく、ユーザとの入出力インターフェースが極めて大きなウェートと占めている。Windows における GUI がそれであり、J 言語はこれら両者を満たした極めて優れた環境といえよう。

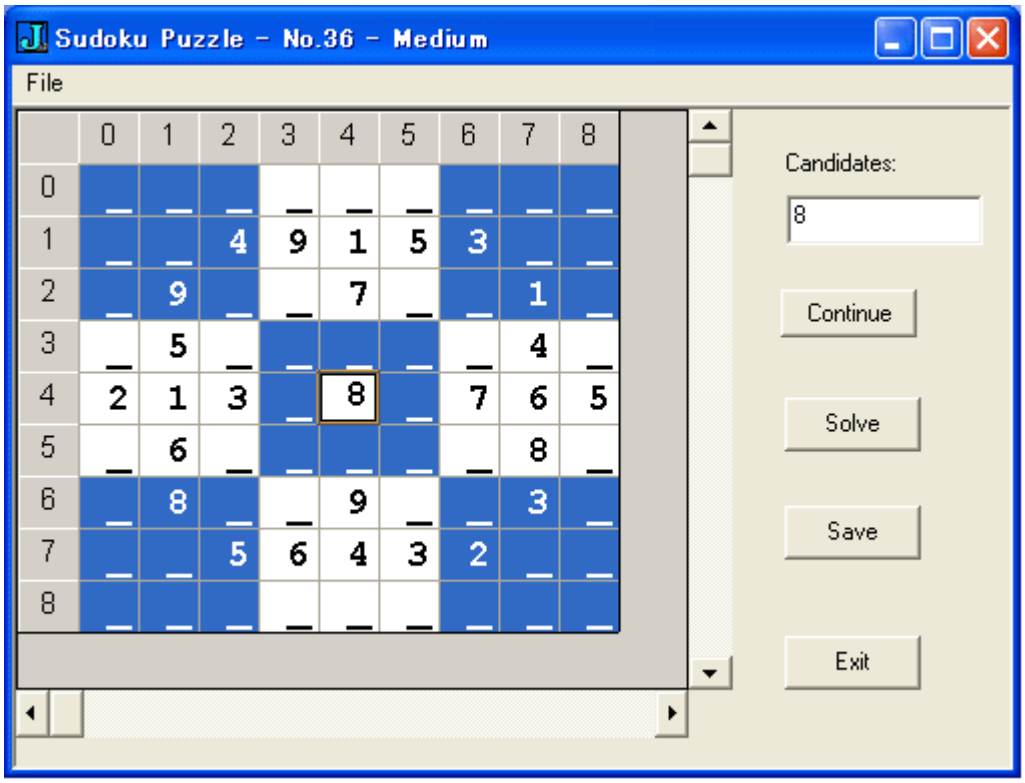
とくに、コンピュータでパズルを楽しむには、単に数理的に計算して解を得るだけでは不十分である。数独パズルでは空白のセルに次々と数字を入れて行くが、画面がスクロールしてしまって、あちこち探すようでは楽しみも半減する。J の Grid はまさに最適で、このためにあるとさえ筆者は思っている。

１．J 数独パズルの改良版

今回の改良では、まず数独のデータベースから簡単にいろいろな問題を取り出せるようにした。起動すると以下のように選択の対話ボックスが開いて、たとえば「No.36 – Medium」の問題[4]が取りだされる。



ごらんのように、Grid のセルの大きさも数独らしくタテヨコ同じにし、文字フォントも大きくした。マウスクリックでセルをクリックすると候補となる数字を教えてくれる機能は以前のとおりである。この例ではセル(4, 4)（0 オリジンで）に数字 8 を入れた場面を示す。Candidates（候補）の数字を参考にしつつ次々と数字を入れて、数独を楽しんだあと、[Continue] ボタンで保存すれば、次回はその場面から続けて遊ぶことも出来る。また、「お手上げになってしまった人」には、[Solve] ボタンでコンピュータが解いてくれる。もちろん、改良版 Hui の数独プログラム[3]がこれをやっていることはいうまでもない。（次ページに示す）



2．J−Grid のプログラム

J のオブジェクト指向の基本については、これまで何回となく述べてきた[1], [2]。プログラムは次の 2 つから成る。

- sudoku2.ijs … メインプログラム、Grid を起動する普通のスクリプト
- psudoku2.ijs … クラスプログラム、数独パズルのための機能を追加

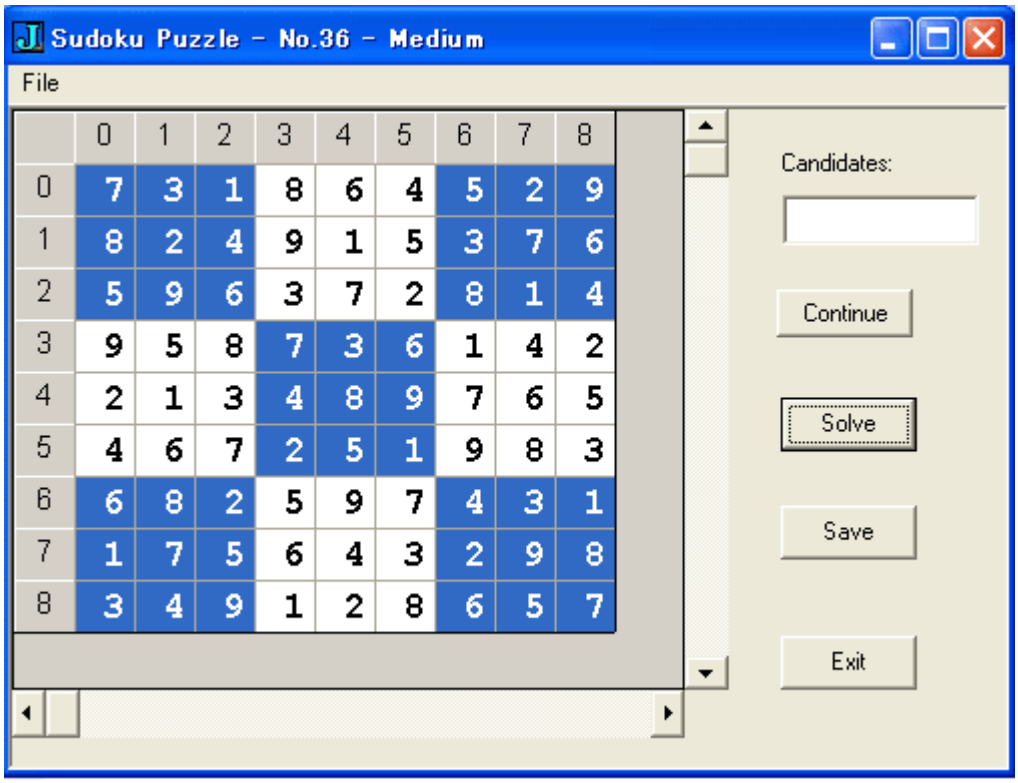
先に発表したプログラム sudoku.ijs, psudoku.ijs では J に常備の汎用クラス・プログラムをほとんどそのまま使用したが、今回はクラス継承の元である jwgrid.ijs, jzgrid.ijs を直接使用して、プログラム起動と同時に数独パズルの画面が直ちに出るようにした。

以下の箇所に修正点があるが、当日、プログラムのコーディング上でコメントしよう。

(1) セルの高さ、幅　　(2) 文字フォント　(3) 数独ブロック範囲の色付け

(4) マウス左クリックによるセル位置

以下は[Solve]ボタンによるパズルの解を示したものである。



これは[Solve]ボタンを押したとき、Hui の数独プログラム[3]を起動させ、その計算結果を Grid 表示するようにしたものである。もちろん、J－Grid はすべて J で書かれているから J の数独プログラムを簡単に呼んで処理できる。

数独パズルではその形からスプレッド・シートの利用はだれでも思いつくが、Excel などでは VBA を用いても、このようなパズルを解くプログラムを作ることは相当至難のわざであろう。J 言語にしてはじめて可能なことと思われる。

文献

[1] 西川利男「J のオブジェクト指向プログラミングーその１、J のＯＯＰとは－簡単な例でやってみる」JAPLA シンポジウム、2005/12/10

[2] 西川利男「J のオブジェクト指向プログラミングーその２、J のスプレッド・シート（Grid）と数独パズルへの適用」JAPLA シンポジウム、2005/12/10

[3] 西川利男「数独(SUDOKU)パズルを J で解く－Labs システムによる Hui のプログラムのトレース」J 言語研究会資料　2006/1/28

[4] 「数独 21」ペンシルパズル本 102（株）ニコリ（2005）

プログラム・リスト

```
NB. Sudoku2.ijs  Sudoku Puzzle ver.2 - by T. Nishikawa, 2006/1/9
NB. Using Class psudoku2.ijs            2006/2/7, 2/12
NB. Using Sudoku Value Database 'suddat.txt', solved by Hui's Program

require 'files'
NB. run '' ==> Various data from database suddat.txt
NB. run 1  ==> Yomiuri data
NB. run _  ==> Continued data
NB. 'Data Name' run 0 ==> New data
run =: 3 : 0
'' run y.
:
corequire 'user¥classes¥psudoku2.ijs'
if. 0 = #x.
 do.
   if. 0 = #y.
     do.
        dat =: open ''
        datn =: DatNam
     else.
        select. y.
          case. 0  do. dat =: 9 9$_ [ datn =: 'New Data'
          case. 1  do. dat =: da_ym [ datn =: 'Yomiuri Data'
          case. _  do. da =: 'm' fread 'sudtmp.txt'
                       datn =: {. da
                       dat =: ". 9 {. }. da
        end.
   end.
 else.
 datn =: x. [ dat =: 9 9$_
end.
w =: 'dat' conew 'psudoku2'
)
```

```
open =: 3 : 0
DAT =: 'm' fread 'suddat.txt'
id =. <. 10%~ {. $DAT
DA =: (10 * i. id){ DAT
DB =: <;._1 ,'/',"1 DA
ix =: > {: wdselect DB
DatNam =: > ix{DB
DC =: ((10*ix) + >: i. 9) { DAT
DatVal =: ". DC -. ' '
)

NB. =====================================================
NB. psudoku2.ijs  class program Sudoku ver. 2
NB.   data input&edit from 'jwatch program'
NB.   solve by Hui's program
NB. 2006/2/12, 2/28(cell R, C from jzgrid, not used sys2cel)
coclass 'psudoku2'
corequire 'jwgrid'
require 'gl2 jzgrid'
require 'files'
PSUDOKU2=: 0 : 0
pc psudoku2;pn "Sudoku Puzzle";
xywh 192 63 34 12;cc save button;cn "Save";
xywh 192 117 34 12;cc cancel button;cn "Exit";
xywh 0 0 168 129;cc grid isigraph ws_border rightmove bottommove;
xywh 0 129 168 11;cc sb scrollbar topmove rightmove bottommove;
xywh 168 0 11 129;cc sbv scrollbarv leftmove rightmove bottommove;
xywh 192 9 50 10;cc label static;cn "Candidates:";
xywh 192 19 50 12;cc cand edit ws_border es_autohscroll;
xywh 192 41 34 12;cc solv button;cn "Solve";
pas 6 6;pcenter;
rem form end;
)
```

```
create=: 3 : 0
wd PSUDOKU2
wdfit ''
formhwnd=: wd'qhwndp'
NB. initialize form here
grid =: '' conew 'jwgrid'
sizeenable__grid=: 1
init__grid 'dat_base_';'grid';'scrollbar';'scrollbarv'
new_flag =: 0
D =: dat_base_
wd 'pn *Sudoku Puzzle - ', datn_base_
('Row';'Col') =: $D
gridws=: Row$50
gridhs=: Col$20
setincwh__grid 0 _20  NB. decrement cell_width
setincwh__grid 6 0    NB. increment cell_height
editenable__grid=: 0
setcolor ''                         NB. color blocked cell
glgridfont0 '"courier new" 22 bold'
wd 'pshow;'
)

NB. coloring blocked cells using by type-2 attribute
setcolor =: 3 : 0
glgridrchw 1 1 3 3
glgridtype 9#2
glgridrchw 1 7 3 3
glgridtype 9#2
glgridrchw 7 1 3 3
glgridtype 9#2
glgridrchw 7 7 3 3
glgridtype 9#2
glgridrchw 4 4 3 3
glgridtype 9#2
)
```

```
destroy=: 3 : 0
wd'pclose'
codestroy''
)
psudoku2_open_button=: 3 : 0
smoutput D
)

psudoku2_exit_button=: destroy
psudoku2_cancel=:psudoku2_cancel_button=:psudoku2_close=:destroy

formselect=: 3 : 'wd''psel '',formhwnd'


psudoku2_sb_button=: 3 : 0
scrollbar__grid sb
wd 'setfocus grid'
)

psudoku2_sbv_button=: 3 : 0
scrollbarv__grid sbv
wd 'setfocus grid'
)

psudoku2_grid_size=:    3 : 'size__grid 0'

psudoku2_grid_mmove=:   3 : 'mmove__grid sysdata'
psudoku2_grid_mblup=:   3 : 'mblup__grid sysdata'
psudoku2_grid_mbldbl=:  3 : 'mbldbl__grid sysdata'

psudoku2_grid_copy=:    3 : 'copy__grid 0'
psudoku2_grid_paste=:   3 : 'paste__grid 0'

psudoku2_ectrl_fkey=:   3 : 0
editenable__grid=: -.editenable__grid
)
```

```
psudoku2_grid_char=:    3 : 0
char__grid sysdata
setcolor ''
glpaintx ''
)


psudoku2_grid_mbldown=: 3 : 0
mbldown__grid sysdata
NB. mouse__grid sysdata
RC =: mark__grid                NB. marked R, C
'R C'=. RC                              NB.  ...
NB. smoutput <: R, C
cdata =: glgridgettext R, C NB. get value
NB. smoutput cdata

NB. ---------------------------------------------
NB. Get Cadidates from Cell Data
NB. cand =: 3 : 0
NB. RC =. y.
NB. smoutput RC
CD =. ": >:i.9
NB. horizontal check ====
cdx =. ''
j=. 1
while. j <: Col do.
  Cel_RC =. R, j
  cdx =. cdx, (glgridgettext Cel_RC), ','
  j =. j + 1
end.
CDX =. ({:cdx) -. '_'
CD =. CD -. CDX
NB. wd 'set e4 ',  ({: ,(":,. CDX),"1 ',')
NB. smoutput CD
```

```
NB. vertical check ====
cdy =. ''
i=.1
while. i <: Row do.
  Cel_RC =. i, C
  cdy =. cdy, (glgridgettext Cel_RC), ','
  i =. i + 1
end.
CDY =. (}:cdy) -. '_'
CD =. CD -. CDY
NB. wd 'set e5 ',  (}: ,(":,. CDY),"1 ',')
NB. smoutput CD

NB. block check ===
cdz =. ''
R0 =. 3*<.3%~_1+R
C0 =. 3*<.3%~_1+C
i=. 1
while. i<:3 do.
  j=. 1
  while. j<:3 do.
    Cel_RC =. (R0+i), (C0+j)
    cdz =. cdz, (glgridgettext Cel_RC), ','
    j =. j + 1
  end.
  i =. i + 1
end.
CDZ =. (}:cdz) -. '_'
CD =. CD -. CDZ
NB. wd 'set e6 ',  (}: ,(":,. CDZ),"1 ',')
NB.  smoutput CD
NB.  smoutput $CD
NB.  smoutput hdump CD
wd 'set cand ',  }: , (,.CD -. ' '),"1 ','

NB. Input Value in Cell --------------------------------
```

```
if. '_' = cdata do.
  editenable__grid =: 1
  mbldown__grid sysdata
  setcolor ''
  glpaintx ''
end.
)
NB. ===============================================
psudoku2_cont_button=: 3 : 0
cd =. ''
i=. 1
while. i <: Row do.
  j=. 1
  while. j <: Col do.
  cd =. cd, (". glgridgettext i, j)
  j =. j + 1
  end.
i =. i + 1
end.
datn_base_ fwrites 'sudtmp.txt'
NB. smoutput (": (Row, Col)$cd)
(": (Row, Col)$cd) fappends 'sudtmp.txt'
)

psudoku2_save_button=: 3 : 0
cd =. ''
i=. 1
while. i <: Row do.
  j=. 1
  while. j <: Col do.
  cd =. cd, (". glgridgettext i, j)
  j =. j + 1
  end.
i =. i + 1
end.
datn_base_ fappends 'suddat.txt'
```

```
NB. smoutput (Row, Col)$cd
(": (Row, Col)$cd) fappends 'suddat.txt'
)

psudoku2_solv_button=: 3 : 0
S =. h2n , sudoku , n2h D
('Row';'Col') =: $S
i =. 1
while. i<:9 do.
  j =. 1
  while. j<:9 do.
   glgridrchw i, j, 1, 1                NB. write value
   glgridtext ": (< <: i, j){S          NB.  ...
   glpaintx ''
   j =. j + 1
  end.
  i =. i + 1
end.
)
=========================================================================
NB Sudoku Data Conversion
subs=: [. & ((((e.&) ((# i.@#)@)) (@])) })
NB. substitute 'x' to 'a' in 'xbbcxda'
NB.   'a' subs 'x' 'xbbcxda'
NB. abbcada
NB.   9 subs 2 (2 1 3 2 4)
NB. 9 1 3 9 4

NB. convert data from Nishikawa's format to Hui's format
n2h =: 3 : 0
9 9$0 subs _ (, y.)
)
NB. convert data from Hui's format to Nishikawa's format
h2n =: 3 : 0
9 9$_ subs 0 (, y.)
)
```

```
NB. ========================================================================
NB. Hui's Sudoku Solving Program
NB. Modified for J3, J4 by T. Nishikawa, 2006/1/2

j       =: (]/. i.@#) ,{;~3#i.3
r       =: 9#i.9 9
c       =: 81$|:i.9 9
b       =: (,j{9#i.9) { j
I       =: ~."1 r,.c,.b
R       =: j, (,|:)i.9 9

regions =: R"_ {"_ 1 ]
free    =: 0&= > (1+i.9)"_ e."1 I&{
ok      =: (27 9$1)"_ -:"2 (0&= +. ~:"1)@regions

ac      =: +/ .*&(1+i.9) * 1: = +/"1

Ip      =: # i.@#       NB. I.(indices) is defined as Ip
ar      =: 3 : 0
 m=. 1=+/"2 R{y.
 j =. Ip +. /"1 m
 k =: 1 i."1~ j{m
 i =: ,(k{"_1 |:"2 (j{R){y.) #"1 j{R
 (1+k) i}81$0
)

assign  =: (+ (ac >. ar)@free)^:_"1

guessa  =: 3 : 0
 if. -. 0 e. y. do. ,:y. return. end.
 b =. free y.
 i =. (i.<./) (+/"1 b){10,}.i.10
 y. +"1 (1+ Ip i{b)*/i=i.81
)
guess   =: ; @: (<@guessa"1)
sudoku  =: guess @: (ok #]) @: assign ^:_ @ ,
```