

J の O O P と G r i d プログラミングーその 5

J による数独パズルー棋譜の自動記録と棋譜データによる実行

西川 利男

0. はじめに

数独パズルの解は、Hui による J のプログラムで、(少なくともユニークな解を持つものについては) きわめて迅速に求めることが出来る。

しかし、真の数独パズルの愛好者、中野嘉弘氏、山下紀幸氏に言わせれば、やはり自分の頭で考え、人手で解かなくては面白みがないと言う。たしかにそうかも知れない。ただ、入力途中経過つまりパズルの棋譜を一々記録しつつ行わないと再現性が保たれず困っているが、これが中々わずらわしいとのことであった。

筆者はそれでは棋譜を自動的に記録するようになれば良いでないか、とは言ったものの、このプログラムが案外面倒である。数独を解くのであれば、(Hui のプログラムがそうだが)、J の通常のプログラムだけで行えるが、この場合はそうはいかない。

数独の盤面上でタテ・ヨコ・ブロックの数字をにらみながら、候補となる数字を埋めて行く際には、ユーザはそのセルの位置を意識しないで、自動的に記録してほしい。このためにはどうしても、ウィンドウズ表示でグリッド入力のユーザ・インターフェースが不可欠であり、そのためのプログラミングが必要になる。

これまで、筆者は J の G r i d による数独パズルを以下のように報告して来た。

その 1 J の O O P とはー簡単な例でやってみる JAPLA シンポジウム 2005/12/10

その 2 J のスプレッドシートと数独パズルへの適用 JAPLA シンポジウム 2005/12/10

その 3 J の Grid でカレンダーを作って遊ぶ J 研究会資料 2006/2/25

その 4 J - Grid による数独パズルをもっと使いやすく J 研究会資料 2006/3/25

今回はそれに手を加えて、棋譜の自動記録と棋譜データによる実行のプログラムを完成することが出来た。プログラムの全体 (sudoku3.ijs, psudoku3.ijs) はかなり大きいので、関係する部分のみ (ほとんどはクラスプログラム psudoku3.ijs) についての説明を行う。

1. マニュアル入力時の棋譜の自動記録

1. 1 マウスクリック時のセル位置の取得

```
psudoku3_grid_mbltdown=: 3 : 0
```

```
mbltdown__grid sysdata
```

```
RC =: mark__grid
```

```
'R C'=: RC
```

```
)
```

ウィンドウズ画面でマウスクリックするとグローバル値 `sysdata` として画面上のピクセル座標値が得られる。これを元に J のクラスプログラム `grid` からの継承関数 `mbldown` と `mark` とにより、グリッドのセルの行、列の値が計算され `R C` を得る。

1. 2 キー入力の値の取得

まず、棋譜記録を ON/OFF する選択を行うラジオボタン `rec` を作る。棋譜記録のファイル `sudmov.txt` をリセットして、棋譜データ `MOV` を設定する。なおデータ `MOV` は文字列

```
ldata (R, C) <=> セル値 (行, 列)
```

のように設定してある。

```
psudoku3_rec_button=: 3 : 0
```

```
wd 'set rec 1'
```

```
ferase 'sudmov.txt'
```

```
MOV =: 1 0$"
```

```
mvi =: 1
```

```
)
```

次にキー入力操作の関数を定義する。ラジオボタン `rec` が ON のときは、セル `R`、`C` の位置から `gl2` の関数 `glgridgettext` によりキー入力した値を得て、`R`、`C` の値とともに `movdata` とする。これをキー入力時で蓄積して `MOV` データとし、同時にファイル `sudmov.txt` に書き込む。なお、キー入力時には文字だけではなく `CR` コードも入れられるのでそれを取り除く処理が必要である。

```
psudoku3_grid_char =: 3 : 0
```

```
char__grid sysdata
```

```
setcolor "
```

```
glpaintx "
```

```
if. 1 = ". rec do. NB. if rec_radiobutton ON => save
```

```
ldata =: glgridgettext R, C NB. get cell_data from R, C
```

```
movdata =: cdata, '(', (':R), ',', (':C), ')' NB. cdata, R, C
```

```
NB. Save cdata except " for CR
```

```
if. ('~' ~: { . movdata) do. MOV =: MOV, movdata end.
```

```
if. ('~' ~: { . movdata) do. smoutput movdata end.
```

```
if. ('~' ~: { . movdata) do. movdata fappends 'sudmov.txt' end.
```

```
end.
```

2. 棋譜データによるステップ実行

棋譜によるステップ実行は次のように行われる。ボタン `mov` が押されたときには、あらかじめ記録された棋譜データ `MOV` を元に指定されたセル位置に値が入れられる。つまり、ボタンを押す毎にワンステップずつ数独パズルが実行される。

```
psudoku3_mov_button=: 3 : 0
if. MOVEND = mvi do. return. end.
MV =. mvi{MOV
NB. smoutput MV
T =. 0{MV
i =. ". 3{MV
j =. ". 5{MV
glgridrchw i, j, 1, 1
glgridtext T
glpaintx "
mvi =: mvi + 1
)
```

3. 数独セルのタテヨコインデックスの1-オリジンへの変更

```
celldata=: 3 :;("each y.),each 0{a.'
ij_oneorigin =: 3 : 0
glgridrchw 0 1 1 9
glgridtext celldata 1 + i.9
glgridrchw 1 0 9 1
glgridtext celldata 1 + i.9
)
```

4. プログラムファイルと関連データファイル

- ・数独メインプログラム `sudoku3.ijs`
- ・数独クラスプログラム `psudoku3.ijs`
- ・数独データファイル `suddat.txt`
- ・棋譜データファイル `sudmov.txt`

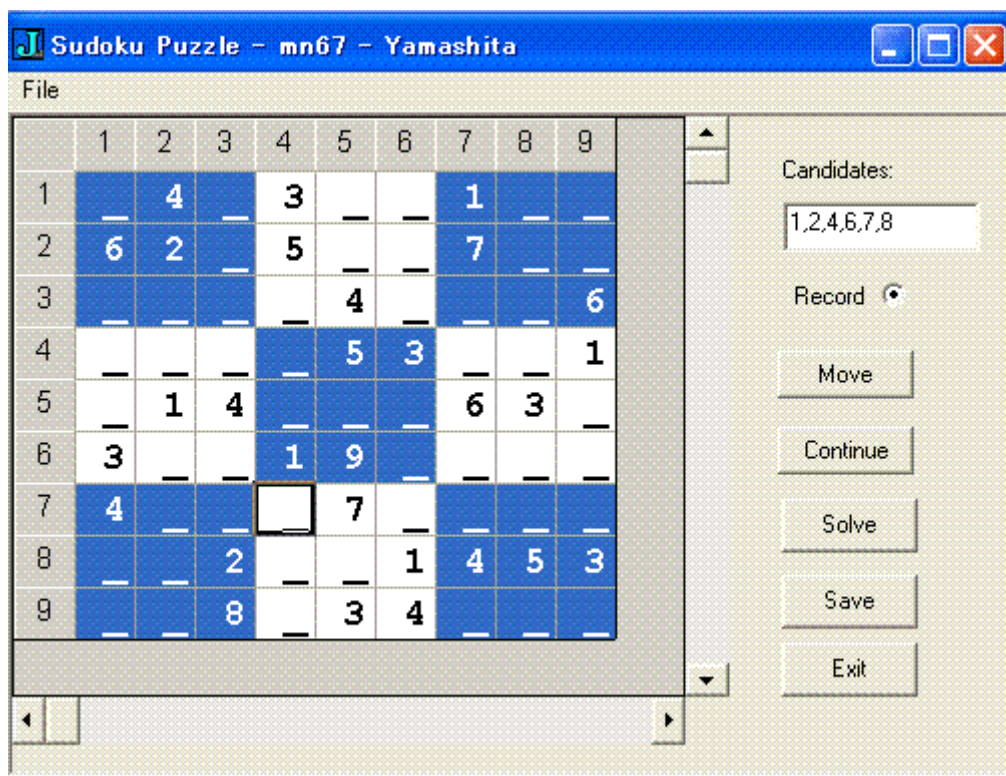
5. J数独パズル (ver. 3) の実際の操作と実行例

5. 1 棋譜の自動記録の実際

```
run 2
```

`mn67` (毎日新聞 6/7 の数独問題) を起動する。

そして、ラジオボタン Record をONにして次々とキー入力する。



すると、キー入力のたびにセルのタテ・ヨコの値も含めて以下のように打ちだされ確認される。

Move: 4 (1, 2)

Move: 3 (4, 6)

Move: 3 (9, 5)

Move: 4 (8, 7)

Move: 1 (6, 4)

ここまで、キー入力したときの数独の盤面は上のようなになる。また

move “

のコマンドにより、保存した棋譜ファイル sudmov.txt の内容を表示することができる。

5. 2 棋譜データによるステップ実行

2 run 2

mn67 (毎日新聞 6/7 の数独問題) に棋譜データを含めて起動する。

ボタン Move を押すたびに、1 ステップずつ自動的に実行されて、解が求まる。

なお、毎日新聞 6/7 の数独問題と棋譜データについては、山下紀幸氏より (FAX 2006/6/9) よりご提供を受けた。ここにお礼申しあげる。

Jによる数独パズルのプログラム・リスト

NB. Sudoku Puzzle ver.3 - by T. Nishikawa, 2006/1/9

NB. record move_data and execute stepwise by these data

NB. from request by Y. Nakano and K. Yamashita

NB. Using Class psudoku3.ijs 2006/5/30, 31

NB. Using Sudoku Value Database 'suddat.txt'

NB. Using Sudoku Solution by Hui's Program

NB. 2006/2/7, 2/12

NB. 2006/6/7 record move, and then move by move_file

NB. 2006/6/16 sudoku_data and sudmov_data from files

NB. run " ==> Select sudoku data data from database suddat.txt

NB. run 1 ==> Yomiuri data

NB. run 2 ==> Mainichi data

NB. 2 run 2 ==> Mainichi data with move data

NB. run _ ==> Continued data

NB. 'Data Name' run 0 ==> New data

```
require 'files'
```

```
run =: 3 : 0
```

```
" run y.
```

```
:
```

```
corequire 'user¥classes¥psudoku3.ijs'
```

```
if. 0 = #y.
```

```
do. NB. Open by Selection Box
```

```
dat =: open "
```

```
datn =: DatNam
```

```
else. NB. Open by Right Argument
```

```
select. y.
```

```
case. 0 do. dat =: 9 9$ _ [ datn =: 'New Data'
```

```
case. 1 NB. Yomouri data
```

```
do. da =: 'm' fread 'suddat.txt'
```

```
datn =: { da [ dat =: ". 9 { . }. da
```

```
case. 2 NB. Maiichi data
```

```

do. da =: 'm' fread 'suddat.txt'
    da =: 60 }. da
    datn =: {. da [ dat =: ". 9 {. }. da
NB. read Move data (mn67mov.txt)
if. 0 <#x. do. movda =: 'm' fread 'mn67mov.txt' end.
case. _ NB. Continued data
do. da =: 'm' fread 'sudtmp.txt'
    datn =: {. da [ dat =: ". 9 {. }. da
end.
end.
w =: 'dat' conew 'psudoku3'
)

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

move =: 3 : 0
fview 'sudmov.txt'
)

```

NB. Sudoku ver. 3

NB. record data of moves when keyboard input

NB. move by reading file 'sudmov.txt' 2006/6/7

```
coclass 'psudoku3'
```

```
corequire 'jwgrid'
```

```
require 'gl2 jzgrid'
```

```
require 'files'
```

```
PSUDOKU3=: 0 : 0
```

```
pc psudoku3;pn "Sudoku Puzzle";
```

```
menupop "File";
```

```
menu new "&New" "" "" "";
```

```
menu open "&Open" "" "" "";
```

```
menu test "&Test" "" "" "";
```

```
menusep ;
```

```
menu exit "&Exit" "" "" "";
```

```
menupopz;
```

```
xywh 192 102 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 85 34 12;cc solv button;cn "Solve";
```

```
xywh 191 69 34 11;cc cont button;cn "Continue";
```

```
xywh 191 52 34 11;cc mov button;cn "Move";
```

```
xywh 194 35 29 11;cc rec radiobutton bs_lefttext;cn "Record";
```

```
pas 6 6;pcenter;
```

```
rem form end;
```

```
)
```

```
create=: 3 : 0
```

```
wd PSUDOKU3
```

```
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
ij_oneorigin " NB. ij_index to 1-origin
setcolor " NB. color blocked cell
glgridfont0 "courier new" 22 bold'
wd 'pshow;'
wd 'set rec 0'
NB. read movdata
MOV =: movda_base_
MOVEND =: { $MOV
mvi =: 0
)

```

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

```


)

psudoku3_open_button=: 3 : 0

smoutput D

)

NB. ij_index to 1-oringin 2006/6/9

celldata=: 3 :;("each y.),each 0{a.'

ij_oneorigin =: 3 : 0

glgridrchw 0 1 1 9

glgridtext celldata 1 + i.9

glgridrchw 1 0 9 1

glgridtext celldata 1 + i.9

)

psudoku3_test_button=: 3 : 0

smoutput R, C

)

psudoku3_exit_button=: destroy

destroy=: 3 : 0

wd'pclose'

codestroy"

)

psudoku3_cancel=:psudoku3_cancel_button=:psudoku3_close=:destroy

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

psudoku3_sb_button=: 3 : 0

scrollbar__grid sb

wd 'setfocus grid'

)

psudoku3_sbv_button=: 3 : 0

```

scrollbarv__grid sbv
wd 'setfocus grid'
)

```

```

psudoku3_grid_size=: 3 : 'size__grid 0'

```

```

psudoku3_grid_mmove=: 3 : 'mmove__grid sysdata'
psudoku3_grid_mblup=: 3 : 'mblup__grid sysdata'
psudoku3_grid_mbldbl=: 3 : 'mbldbl__grid sysdata'

```

```

psudoku3_grid_copy=: 3 : 'copy__grid 0'
psudoku3_grid_paste=: 3 : 'paste__grid 0'

```

```

psudoku3_ectrl_fkey=: 3 : 0
editenable__grid=: -.editenable__grid
)

```

NB. select radiobutton ON/OFF for saving movdata into file

```

psudoku3_rec_button=: 3 : 0
wd 'set rec 1'
ferase 'sudmov.txt'
MOV =: 1 0$"
mvi =: 1
)

```

```

psudoku3_grid_char=: 3 : 0
char__grid sysdata
setcolor "
glpaintx "

```

if. 1 = ". rec do. NB. if rec_radiobutton ON => save movdata

```

  cdata =: glgridgettext R, C NB. get cell_data from R, C
  movdata =: cdata, '(', ("R), ',', ("C), ')' NB. C_val(R, C)
  if. (' ~: {. movdata) do. MOV =: MOV, movdata end.
  if. (' ~: {. movdata) do. smoutput movdata end.
  if. (' ~: {. movdata) do. movdata fappends 'sudmov.txt' end.
end.

```

)

NB. step go from MOV data

psudoku3_mov_button=: 3 : 0

if. MOVEND = mvi do. return. end.

MV =. mvi{MOV

NB. smoutput MV

T =. 0{MV

i =. ". 3{MV

j =. ". 5{MV

glgridrchw i, j, 1, 1

glgridtext T

glpaintx "

mvi =: mvi + 1

)

NB. Start of Function =====

psudoku3_grid_mbltdown=: 3 : 0

mbltdown__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. -----

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.

```

```

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
    mblldown__grid sysdata
    setcolor "
    glpaintx "
end.
)
NB. End of Function =psudoku3_grid_mblldown=====

NB. =====
psudoku3_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'
)

```

```

psudoku3_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'
)

```

```

psudoku3_solv_button=: 3 : 0
S =. h2n , sudoku , n2h D
('Row';'Col') =: $S
i =. 1
while. i <: 9 do.
    j =. 1
    while. j <: 9 do.
        glgridrhw i, j, 1, 1      NB. write value
        glgridtext " (< <: i, j){S  NB. ...
        glpaintx "
        j =. j + 1
    end.
    i =. i + 1
end.
)

```

```

psudoku3_new_button=: 3 : 0
new_flag =: 1
D =: 9 9$_

```

```

init__grid 'D';'grid';'scrollbar';'scrollbarv'
('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'
i =. 1
while. i<:9 do.
  j =. 1
  while. j<:9 do.
    glgridrchw i, j, 1, 1 NB. write blank
    glgridtext ":_ NB. ...
    glpaintx "
    j =. j + 1
  end.
  i =. i + 1
end.
)

```

NB. Nishiawa's Utilities =====

```
wr=. 1!:2&2
```

```
each=: &. >
```

```
deb=: #~ (+. 1&|. @(> </¥))@(' '&~:)
```

```
h=. '0123456789ABCDEF'
```

```
dfh=: 16&#. @ (h&i.) f.
```

```
hfd=: dfh ^:_1 f.
```

NB. hex (adverb)

NB. e.g. 'FF' + hex '8'

```
hex=: &. dfh
```

```
val=: a. & i.  
chr=: val ^: _1  
hdump=: }:@,@("1' "_ )@hfd@val f.
```

=====
=====
NB Sudoku Data Conversion

```
subs=: [. & (((e.&) ((# i.@#)@)) (@)) }
```

NB. substitute 'x' to 'a' in 'xbbexda'

NB. 'a' subs 'x' 'xbbexda'

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. 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=.81
```

```
)
```

```
guess   =: ; @: (<@guessa"1)
```

```
sudoku  =: guess @: (ok #) @: assign ^:_" @ ,
```