Manual for `sgame.sty` **and** `sgamevar.sty`
by
Martin J. Osborne
`Martin.Osborne@utoronto.ca`
For Version 2.15, 2013-6-24

1. Introduction

`sgame.sty` and `sgamevar.sty` are LaTeX 2$_\varepsilon$ style files for drawing strategic games. The latest versions are available at

> `http://www.economics.utoronto.ca/osborne/latex.`

The style `sgamevar.sty` (but not `sgame.sty`) is compatible with `beamer`. Both styles are incompatible with the `memoir` class and with `tablularx.sty` and `array.sty` (and hence any package, like `jurabib.sty`, that loads `array.sty`). Please notify me of bugs.

2. sgame.sty versus sgamevar.sty

`sgame.sty` and `sgamevar.sty` differ in two respects.

- The character that "tabs" to a new column is `&` in `sgame.sty` and `\>` in `sgamevar.sty`. In every other respect the syntax is identical. This manual applies to `sgame.sty`. If you are using `sgamevar.sty`, replace every subsequent occurrence of `&` with `\>`.

- `sgame.sty` produces a game environment that cannot be used as an argument of a macro, whereas `sgamevar.sty` does not have this limitation. You need to use `sgamevar.sty`, for example, to be able to make a game environment an argument of TeXPower's `\stepwise` macro.

(I prefer the `&` syntax, but have been unable to find a way of implementing it without changing the `\catcode` of `&`, which prevents a game environment from being the argument of a macro.)

3. Installation

- Put `sgame.sty` (and/or `sgamevar.sty`) in a directory from which TeX reads input files. (In MiKTeX the directory might be something like `\miktex\localtexmf\tex\latex\` or a subdirectory thereof.)

- Let TEX know that sgame and/or sgamevar have arrived. (In MiKTeX, "refresh the filename database".)

- To use the package in a document, put the line \usepackage{sgame} or \usepackage{sgamevar} in the preamble.

4. Compatibility with other packages

- The style is incompatible with colortbl.sty.

- In beamer, use sgamevar (or put a game in a \begin{frame}[fragile]...\end{frame} environment).

- In the case of incompatibilities with other styles, try using sgamevar rather than sgame.

5. Description

The style defines two environments, game and game*. Each environment can be used in four ways, depending on the values of its optional arguments. In each case, the starred version differs from the unstarred version in that action labels are not accommodated.

5.1 No optional argument

Without any optional arguments, the syntax for a game is

\begin{game}{*number-of-rows*}{*number-of-columns*}*game-body*\end{game}

where *number-of-rows* is the number of rows in the game and *number-of-columns* is the number of columns. *game-body* consists of rows in the style of the rows in a tabular environment, & "tabbing" to a new cell, and \\ ending a row.

The following example is illustrated in Figure 1.

```
\def\sgtextcolor{blue}%
\def\sglinecolor{red}%
\begin{figure}[htb]\hspace*{\fill}%
\begin{game}{2}{2}
    & $L$   & $R$ \\
$T$ & $2,2$ & $100,35$ \\
```

2

```
$B$ & $3,0$ & $350,16$
\end{game}\hspace*{\fill}%
\begin{game*}{2}{2}
$2,2$ & $100,35$ \\
$3,0$ & $350,16$
\end{game*}\hspace*{\fill}%
\caption[]{A two row, two column strategic game, with and without
action labels.}
\end{figure}
```

	L	R
T	$2,2$	$100,35$
B	$3,0$	$350,16$

$2,2$	$100,35$
$3,0$	$350,16$

Figure 1. A two row, two column strategic game, with and without action labels.

- Note that the width of every column is the same and the content of each cell is centered vertically.

- As of version 2.15, spaces on either side of the text in each cell are ignored, so you can add spaces to your input file to make it easier to read (or just for the heck of it, for that matter).[1]

- The first & **must** be preceded by a space token (space or carriage return). (It may not immediately follow the closing brace of the second argument.)

- If all the text in every cell of the payoff matrix is in math mode, you can avoid typing all the $'s by setting \gamemathtrue, either by putting this command in your file, or by changing \gamemathfalse to \gamemathtrue in the style file. Under this setting, all text in every cell of the payoff matrix is automatically in math mode. (Note that even if the text in the cells consists only of comma-separated pairs of numbers, it should still be in math mode, else the spacing is not correct.)

---

[1]The claim in previous versions of this manual that this behavior held for earlier versions of the style file is incorrect.

5.2 One optional argument: game label

With one optional argument, the syntax of the game environment is

\begin{game}{*number-of-rows*}{*number-of-columns*}[*game-label*]*game-body*
\end{game}

where *number-of-rows* is the number of rows in the game, *number-of-columns* is the number of columns, and *game-label* is a label for the game that is centered beneath the game. (The sense in which it is centered may be changed by setting \irpawcgltrue or \irplwcgltrue—see Section 12.) This case is useful if two or more games are positioned side-by-side, and each requires a label.

The following example is illustrative.

```
\begin{figure}[htb]\hspace*{\fill}%
\begin{game}{2}{2}[$A$]
      & $L$       & $R$\\
$a$    &&$2,2,1$   &&$0,3,0$\\
$b$    &&$3,0,2$   &&$1,1,4$
\end{game}\hspace*{20mm}%
\begin{game}{2}{2}[$B$]
      & $L$       & $R$\\
$a$    &&$2,3,0$   &&$0,4,1$\\
$b$    &&$3,1,2$   &&$1,2,0$
\end{game}\hspace*{\fill}%
\caption[]{A three player strategic game, in which
player~3 chooses $A$ or $B$.}
\end{figure}
```


	L	R			L	R
a	2,2,1	0,3,0		a	2,3,0	0,4,1
b	3,0,2	1,1,4		b	3,1,2	1,2,0
	A				B	

Figure 2. A three player strategic game, in which player 3 chooses *A* or *B*.

5.3 Two optional arguments: player labels

In the case of two optional arguments, the syntax of the `game` environment is

\begin{game}{*number-of-rows*}{*number-of-columns*}[*row-player-label*]
[*column-player-label*]*game-body*\end{game}

where *number-of-rows* is the number of rows in the game, *number-of-columns* is the number of columns, and *row-player-label* and *column-player-label* are the names of the players.

The following example illustrates this case.

```
\begin{figure}[htb]\hspace*{\fill}%
\begin{game}{2}{3}[Player~1][Player~2]
      & $L$    & $M$    & $R$\\
$T$   &$2,2$   &$2,0$   &$0,3$\\
$B$   &$3,0$   &$0,9$   &$1,1$
\end{game}\hspace*{\fill}%
\caption[]{A two row, three column strategic game with
player labels.}
\end{figure}
```

	Player 2		
	L	*M*	*R*
Player 1 *T*	2,2	2,0	0,3
B	3,0	0,9	1,1

Figure 3. A two row, three column strategic game with player labels.

5.4 Three optional arguments: player labels and game label

The case of three optional arguments combines the two previous cases: the syntax of the `game` environment is

\begin{game}{*number-of-rows*}{*number-of-columns*}[*row-player-label*]
[*column-player-label*][*game-label*]*game-body*\end{game}

where *number-of-rows* is the number of rows in the game, *number-of-columns* is the number of columns, *row-player-label* and *column-player-label* are the names of the players, and *game-label* is a label for the game.

The following example illustrates this case.

```
\begin{figure}[htb]\hspace*{\fill}%
\begin{game}{2}{3}[Player~1][Player~2][$A$]
      & $L$    & $M$    & $R$\\
$T$    &$2,2$   &$2,0$   &$0,3$\\
$B$    &$3,0$   &$0,9$   &$1,1$
\end{game}\hspace*{\fill}%
\caption[]{A two row, three column strategic game with
player labels and a label.}
\end{figure}
```
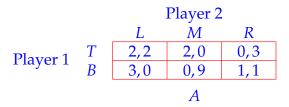

Figure 4. A two row, three column strategic game with player labels and a label.

6. Cell formatting

You may regard the format of the payoffs in the first column of the game in Figure 5 as unsatisfactory, and the formatting in the left panel of Figure 6 as superior. This latter formatting is implemented by adding "phantom" minus signs before the "2" in the top left cell and before the "3" in the bottom left cell. The top left cell, for example, is entered as `$-2,\phantom{-}2$`. (`\phantom{<x>}` leaves a space whose size is equal to the size of the object `<x>`.)

Figure 5. A strategic game.

You may want to consider other formats. The right panel of Figure 6 shows an alternative. This format is achieved be entering the element of each cell as

```
\begin{array}{c}#1\\[1.6mm]#2\end{array},
```

where #1 is player 1's payoff and #2 is player 2's payoff. (Actually, I defined the macro

```
\def\stackedpayoffs#1#2{%
  \begin{array}{c}#1\\[1.6mm]#2\end{array}
}
```

and entered each element as `\stackedpayoffs{#1}{#2}`. I set `\gamestretch` to be 2.1 to improve the appearance of the game.)

Figure 6. Two alternative presentations of the game in Figure 5.

Figure 7 shows two more alternatives. The one on the left is achieved by entering the element of each cell as

```
\begin{array}{rl}#1&\\[1.6mm]&#2\end{array},
```

and the one on the left differs only in that there is a `\phantom{-}` in front of player 2's payoff in the top left cell, player 1's payoff in the bottom left cell, and player 2's payoffs in both of the right cells.

7.   Vertical position of text within cells

The vertical distance between the lines at the top and bottom of a row in LaTeX's `tabular` environment is equal to `\arraystretch` times `\baselineskip`. The baseline of the text in each cell is placed 30% of the distance from the bottom line. Thus if `\arraystretch` is large relative to the height of the text, the text is not near the middle of the row. (See the left panel of Figure 8; or try setting `\arraystretch` equal to 5 and process a document with a `tabular` environment.) (See the definitions of `\strutbox` and of `\@array` in `latex.ltx`.)

sgame.sty makes the vertical distance between the lines at the top and bottom of a row of a game the same (`\arraystretch` times `\baselineskip`),
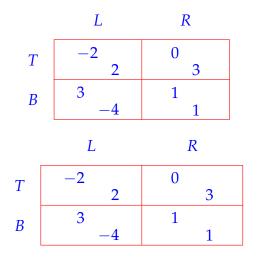
Figure 7. Two additional presentations of the game in Figure 5.

but places the baseline of the text in each cell so that the distances above and below an upper case I placed on the baseline are equal. (See the right panel of Figure 8.) (If you'd like to use another character, define this character to be \sg@alignchar.)
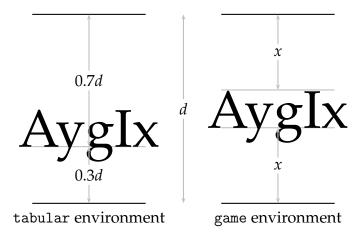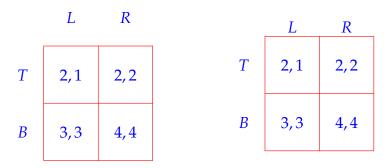

Figure 8. Vertical positioning of text within cells in the tabular and game environments.

If you want to use sgame's vertical positioning in your tabular environments, as well as your game environments, set \gamevaligntrue before the environment. (If you want to switch back at some point, you can do so by setting \gamevalignfalse.)
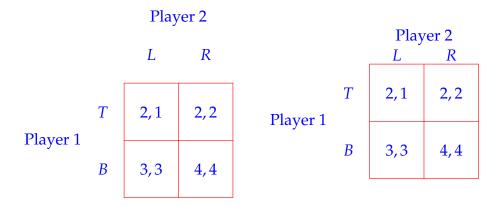
If the height of the text in a cell exceeds \arraystretch times \baselineskip, then the height of the row is increased to accommodate the

text (just as the distance between lines of text is increased to accommodate large text). The result is that the lines at the top and bottom of the row touch the top and bottom of the text, and this row is taller than the others. To improve the appearance of the table you may want to increase \gamestretch (analogous to \arraystretch), to put some white space above and below your oversized text.

When \gamestretch is increased from 1, the height of *every* row, including the first row (the action labels), is increased. If you don't want to increase the space under the action labels, declare \ssualfalse. ("ssual" stands for "stretch space under action labels".) See Figures 10 and 9 for examples (for games with and without player labels).

Figure 9. Positioning the action labels of the column player when gamestretch is larger than 1, for a game without player labels. In the left game we have ssualtrue (the default); in the right game ssualfalse is declared.

Figure 10. Positioning the action labels of the column player when gamestretch is larger than 1, for a game with player labels. In the left game we have ssualtrue (the default); in the right game ssualfalse is declared.

8. Thick lines

To change the width of the cell boundaries, set \arrayrulewidth (by writing something like \arrayrulewidth2mm or, if you prefer, \setlength{\arrayrulewidth}{2mm}). In addition, you may need to stretch the game vertically to accommodate the thicker lines. To do so, set \gamestretch (by writing something like \renewcommand{\gamestretch}{2}). Here's an example.

```
\def\sgtextcolor{blue}%
\def\sglinecolor{red}%
\renewcommand{\gamestretch}{2}
\arrayrulewidth2mm
\begin{figure}[htb]\hspace*{\fill}%
\begin{game}{2}{2}
      & $L$    & $R$\\
$T$   &&$2,2$   &&$100,35$\\
$B$   &&$3,0$   &&$350,16$
\end{game}\hspace*{\fill}%
\caption[]{Thick lines.}
\end{figure}
```

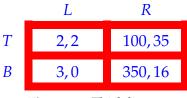|	L	R
T	2,2	100,35
B	3,0	350,16

Figure 11. Thick lines.

9. Crossing out actions

If you know nothing about LaTeX, you may think that creating Figure 12 is easy. If you know a bit about LaTeX, you may think that creating it is impossible. Given PSTricks, the naïf is right: all you need to do is add \usepackage{pstricks,pst-node} to the preamble of your document, define a couple of macros specifying what the strikeout lines should look like, add some node labels to the game, and specify how the nodes should be connected. The only difficulty comes in getting the lengths of the lines right.

I have been unable to get the parameters to work as they are described in Tim van Zandt's manual, but with the help of BaKoMa Word it's easy to fiddle around with them until the lines are the way you want them. The code for this game follows the figure.
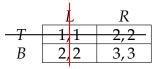
	L	*R*
T	1,1	2,2
B	2,2	3,3

Figure 12. Crossing out actions in a strategic game.

```
\def\sgtextcolor{black}%
\def\sglinecolor{black}%
% define the strikeout lines
\newcommand\strike[2]{%
  \ncline[linewidth=1.2pt,nodesep=8pt]{#1}{#2}}
\newcommand\redStrike[2]{%
  \ncline[linewidth=1.2pt,nodesep=-14pt,linecolor=red]{#1}{#2}}

\begin{game}{2}{2}
                      &\rnode[t]{a12}{$L$}    &$R$\\
\Rnode[href=20]{a21}{$T$} &$1,1$                      &\Rnode{a23}{$2,2$}\\
$B$                   &\rnode[b]{a32}{$2,2$} &$3,3$
\end{game}
% specify the nodes to be connected
\strike{a21}{a23}
\redStrike{a12}{a32}
```

10. Coloring cells

The package is incompatible with `colortbl.sty`. However, here's one way to color cells. It is not elegant: you need to manually coordinate the parameters (which is easy using BaKoMa, but not otherwise). But it works; the output is in Figure 13.

```
\sgcolsep=0.5pt
\renewcommand{\gamestretch}{1.5}
\def\highlight#1{\colorbox{yellow}{#1}}
\fboxsep=6pt
```

11

```
\begin{figure}[htb]\hspace*{\fill}%
\begin{game}{2}{2}
&$L$&$R$\\
$T$\hspace{\fboxsep}&\highlight{$1,1$}&$2,2$\\
$B$\hspace{\fboxsep}&$3,0$&$0,3$\\
\end{game}\hspace*{\fill}%
\caption[]{A game with a colored cell.}\label{f:coloredCell}
\end{figure}
```

	L	R
T	1,1	2,2
B	3,0	0,3

Figure 13. A game with a colored cell.

11. Extra columns and rows

Sometimes you may want to add a column to the right of a game, or a row below it, or both, as in Figure 14. You can do so using the trick of putting the extra text in an \hbox of width zero. If you want to do this a lot, you probably should generate the sizes of the spaces and skips from the data of the table, rather than hard-coding them as I have done.

```
\def\hangUnder#1{%
  \hbox to 0pt{\hspace{-11pt}\raisebox{-8mm}[0pt][0pt]{#1}}%
}
\sgcolsep=5pt
\begin{figure}[htb]\hspace*{\fill}%
\begin{game}{3}{3}
&$L$ (0)%
  &$C$ (${\textstyle\frac{1}{3}}$)%
    &$R$ (${\textstyle\frac{2}{3}}$)\\
$T$ (${\textstyle\frac{3}{4}}$)%
  &$1,2$%
    &$3,3$%
      &$1,1$\hbox to 0pt{\hspace{20pt}${\textstyle\frac{5}{3}}$}\\
```

```
$M$ (0)%
  &$1,2$%
    &$0, 1$%
      &$2, 2$\hbox to 0pt{\hspace{20pt}${\textstyle\frac{4}{3}}$}\\
$B$ (${\textstyle\frac{1}{4}}$)%
  &$1,4$\hangUnder{$\frac{5}{2}$}%
    &$5,1$\hangUnder{$\frac{5}{2}$}%
      &$0,7$\hangUnder{$\frac{5}{2}$}%
          \hbox to 0pt{\hspace{20pt}${\textstyle\frac{5}{3}}$}\\
\end{game}\hspace*{\fill}%
\vspace{8mm}
\caption[]{A game with an extra column and row.}\label{f:extraColumn}
\end{figure}
```

	L (0)	C ($\frac{1}{3}$)	R ($\frac{2}{3}$)	
T ($\frac{3}{4}$)	1,2	3,3	1,1	$\frac{5}{3}$
M (0)	1,2	0,1	2,2	$\frac{4}{3}$
B ($\frac{1}{4}$)	1,4	5,1	0,7	$\frac{5}{3}$
	$\frac{5}{2}$	$\frac{5}{2}$	$\frac{5}{2}$	

Figure 14. A game with an extra column and row.

12. Parameters

\gamestretch: a number that controls the spacing between rows, analogous to \arraystretch. Possible values: positive real numbers. Default: value of \arraystretch when sgame.sty is loaded (1 in standard styles). Example: \renewcommand{\gamestretch}{2}.

\sgcolsep: horizontal padding within cells. Possible values: any dimension. Default: value of \tabcolsep when sgame.sty is loaded. Example: \sgcolsep=10pt.

\sglabelsep: vertical space between game and label. Possible values: any dimension. Default: 5pt. Example: \sglabelsep=10pt.

\irpawcgltrue ("include row player actions when centering game label"): causes game label to be centered under box consisting of row player actions

and payoff matrix. Default: \irpawcglfalse (i.e. game label is centered under box consisting only of payoff matrix). Example: \irpawcgltrue.

\irplwcgltrue ("include row player label when centering game label"): causes game label to be centered under box consisting of row player label, row player actions, and payoff matrix. (Is overridden by \irpawcgltrue.) Default: \irplwcglfalse (i.e. game label is centered under payoff matrix). Example: \irplwcgltrue.

\gamemathtrue: put all text in cells in payoff matrix in math mode. Default: \gamemathfalse. Example: \gamemathtrue.

\gamevaligntrue: vertically align the text within cells of a tabular environment as they are aligned in a game environment. Default: \gamevalignfalse. Example: \gamevaligntrue. (See Section 7.)

\ssualtrue: stretch space under action labels when \gamestretch is greater than 1. Default: \ssualtrue. Example: \ssualfalse. (See Section 7.)

\sglinecolor: color of lines around payoffs. Default: black. Example: \def\sglinecolor{lightgray}.

\sgtextcolor: color of text in game (action labels, payoffs, player labels). Default: black. Example: \def\sgtextcolor{blue}.

13. History

Version 1

1993-12 First version, following a suggestion of Michael Carter.

Version 2

1997-10 Labels allowed (via additional optional argument). Parameters added: \sglabelsep, \sgcolsep, \irplwcgltrue, \irpawcgltrue, and \gamemathtrue.

2002-6 (version 2.1) Modification of LATEX's \@array changed so that it does not change spacing in tabular environment. Color functionality added. Parameters added: \gamevaligntrue, \ssualtrue (allowing more control over vertical spacing when \gamestretch is greater than 1), \sglinecolor, \sgtextcolor.

2004-5-13 (version 2.11)   Minor bug fix.

2004-5-21 (version 2.12)   Minor bug fix.

2005-7-8 (version 2.13)   Minor bug fix.

2008-12-2 (version 2.14)   Minor bug fix: length of bottom line (bug inherited from `tabular` environment; noticeable only when lines are thick). Bug pointed out by Marcel Priebsch.

2010-11-1   Section 9 added to the manual, following a query by a user.

2011-1-14   Section 10 added to the manual.

2013-2-9   Section 11 added to the manual.

2013-6-24 (version 2.15)   1. Longstanding (probably since 1993) bug fix: spaces before and after the text in any cell are ignored. 2. The `color` package is now loaded by the style, and does not need to be loaded explicitly.