

linebreaks can reliably be placed after control words and numerical assignments. We have seen *TeX*'s buffer size exceeded when % was placed after every line.

A wider perspective in the matter of naming macros can prevent problems that occur when definitions are overwritten as articles are run together. The names of control sequences used in plain, L^AT_EX, and A_MS-T_EX are documented and authors should avoid using them for other purposes. It is also wise to avoid commonly used names such as \temp, \result, \i, and \mac in presenting code that might be cribbed by other users. The frequently used technique of temporarily \catcodeing a character to be a letter (e.g. the @) provides a good method of hiding control sequences so that they will not be clobbered later. Readers are in need of small macros to do little tricks, and they often try suggestions brought forth in *TUGboat*. A little extra effort in making these macros consistent with the macros in wide distribution and in making them robust will be much appreciated.

Electronic Documentation and Submission Procedure

In addition to *tugboat.sty*, *ltugboat.sty*, and *tugboat.com*, a copy of this article, *tugguide.tex*, will be available at most *TeX* archives, including those at Clarkson and Aston.

Please address all electronic correspondence to the *TUGboat* maildrop:

TUGboat@Math.AMS.com

Mail to either of our personal addresses is liable to go unseen if vacation or illness intervenes. We also request that you supply an abstract of any expository article. This will be used as the basis for translation of abstracts to languages other than that in which the article is published.

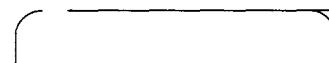
- ◊ Ron Whitney
TeX Users Group
P. O. Box 9506
Providence, RI 02940-9506
TUGboat@Math.AMS.com
- ◊ Barbara Beeton
American Mathematical Society
P. O. Box 6248
Providence, RI 02940-6248
TUGboat@Math.AMS.com

Round boxes for plain *TeX*

Garry Glendown

Doing presentation sheets, I stumbled over a small thing I had been missing for quite a while: boxes. Well, normal boxes are boring, so I thought about doing boxes with round corners.

To do that, I took a look at the circle fonts used for the L^AT_EX pictures. They would work out fine. But, despite of all my *TeX* knowledge and the information from *The TeXbook*, it didn't work. Either the boxes would look like this:



or like this:



or some other, not very encouraging, way. After some hours (I think it was about 2 1/2 or so) I finally solved the problem as found in the listing below.

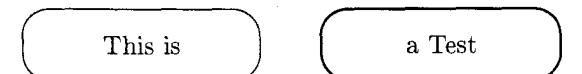
The problem is the strange (at least for normal usage) way the circle font has the width and reference point set. The width is exactly twice as big as the quarter circle, and the reference point of the right two quarters is far beyond the character. So, in order to get the right positioning of the characters, the boxes have to be much wider in the inside than they are on the outside.

Using RBox. To use the RBox-Macro, there are two simple forms: \roundBox and \RoundBox. Both get two parameters: the size of the box as a percent of \hsize, and the text. When calling \roundBox, you will get a box with a border .4pt thick; \RoundBox will result in one with a .8pt border.

If you type

```
\hbox{%
  \roundBox{.4}{This is}%
  \RoundBox{.4}{a Test}%
}
```

it will result in:



In addition to these two 'interface'-macros, you may use the internal function called \RBox. The syntax is the following:

```
\RBox{{total width}} {{width of inner box}}
      {{width of the frame}} {{text}}
```

When using the \RBox-Command, make sure that you define the font '\cf' to be either `circle10` (.4pt thick) or `circlew10` (.8pt thick).

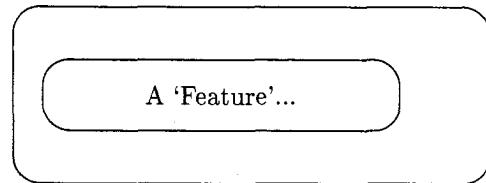
You may use \RBox for some strange effects, like:



This was done by setting the thickness of the normal lines (parameter #3) to 0pt. Or you can do things like this:



Undesirable Features. In the moment, I only know of one little problem: the outside size of the round box is always smaller than the width it is told. So, when putting one box into another, it will look like this:



But I don't see any way out of that problem. If somebody out there has a solution to that problem then please tell me ...

```
%%%%%%% gdround.mac

% \copyright 1989 by Garry Glendown
%           Box R, APO NY 09141 (U.S. Mail)
%   or    G\"uldene Kammer 35,
%         6430 Bad Hersfeld FRG
%         Tel.: (FRG) 06621/77923
%   eMail: Garry@DGIHRZ01.BITNET
%
% \roundBox{hsizen}{Text} thin frame (.4pt)
% \RoundBox{hsizen}{Text} thick frame (.8pt)
% Example:

% internal procedure:
% \RBox#1#2#3#4 where
%   #1 = {outside width}
%   #2 = {inside(text) width}
%   #3 = {width of frame}
%   #4 = {Text}
```

```
% make sure you define the font
% \cf right when using \rbox!
%
% quarter circles:
\def\luff{\cf\char19} % left upper qtr
\def\ruf{\cf\char16} % right upper qtr
\def\llf{\cf\char18} % left lower qtr
\def\rlf{\cf\char17} % right lower qtr
%
\newdimen\wtemp
\newdimen\ww    % width of words
\newdimen\wc    % width of corner
\newdimen\wo    % width of corner + white
\newdimen\wl    % width of leaderbox
%
\def\RBox#1#2#3#4{%
\setbox0=\hbox{\luf}%
\wc=\wd0
\setbox0=\hbox{%
\vrule width#3%
\hbox to #1{%
\hfil%
\#4\hfil%
}%
\vrule width#3%
}%
\ww=\wd0
\wl=\ww
\advance\wl by-#3
\advance\wl by-\wc
\wo=\wc
\divide\wc by 2
\vbox{%
\offinterlineskip%
\hbox{%
\hbox to\wc{\luf\hss}%
\hbox to\wl{%
\leaders
\hrule height#3
\hfil
}%
\hbox to\wc{\hfil}%
\hbox to\wo{\hss\ru}%
}%
\box0
\hbox{%
\hbox to\wc{\llf\hss}%
\hbox to\wl{%
\leaders
\hrule height#3
\hfil
}%
\hbox to\wc{\hfil}%
\hbox to\wo{\hss\rl}%
}}
```

```

}%
}
} % end \def\RBox

\def\roundBox#1#2{%
\wtemp=#1\hsize
\advance\wtemp by-.05\hsize
\font\cf=circle10
\RBox{#1\hsize}{\wtemp}{.4pt}{#2}%
}

\def\RoundBox#1#2{%
\wtemp=#1\hsize
\advance\wtemp by-.05\hsize
\font\cf=circlew10
\RBox{#1\hsize}{\wtemp}{.8pt}{#2}%
}

%
% End of RoundBox
%

```

◊ Garry Glendown
 Box R, APO NY 09141
 or
 Güldene Kammer 35
 6430 Bad Hersfeld
 FRG
 Phone: 0662177923
 Bitnet: Garry@DGIHRZ01

Printing Annotated Chess Literature in Natural Notation

Zalman Rubinstein

There have been several recent attempts to apply the **TEX** and **METAFONT** computer languages to design a chess literature printing package. Appelt [1] suggested a design for printing chessboards, and this author [2] described a simple **METAFONT** chess font. Based on these results the present note introduces a full **TEX** macro which approximates the actual requirements of chess printing, namely:

1. Chess moves are printed in the source file in their natural appearance (as, e.g., Pe2-e4, Nc3-e4) without recourse to control sequence notation (see [1]).
2. Chess moves normally include annotations or comments as to the adjudged value of the move, such as "h7-h6?" for a questionable move, "Bd3-c2!" for a good move. The length of these annotations varies and can include quite a number of symbols, such as +, -, ++, ±, and others.
3. Chess literature can start a game from its natural starting position or from a ready setup position as necessary.
4. The printed form of a move may vary from the natural appearance of the move. For example, Pe2-e4, moving a pawn from square e2 to square e4, is usually denoted by e2-e4 or even by e4 (we shall comment on this at the end of this note).

The **TEX** macros to which I refer satisfy the above requirements by using in various ways the category management ability of the **TEX** language so that the letters K, Q, R, B, N and P denoting the chess pieces play the role of different control sequences at different times. In addition, the end of line character and the paragraph control sequence are redefined for various purposes. The two basic sequences of the macros are:

(a)

```

\ClearBoard
\White Kg1 Qd1 ...
\Black Kg8 Qd8 ...
>ShowBoard (optional)
<text> ...

```

and

(b)

```

\BlacksMove \movecounter8
\StartPlay
Ph7-h6?

```